

Enemy–Victim Interactions

Enemy–victim interactions, a.k.a. consumer–resource, or exploitative interactions are among the most dramatic interactions we can witness, whether that interaction is a cheetah chasing down a gazelle, or an osprey diving for a fish. Humans have always had a fascination with predators and death, and ecologists are humans, for the most part. In addition, plants are consumers too, but watching grass take up nitrate and CO_2 (i.e. grow) is somewhat less scintillating than tracking wolves across the tundra. Nonetheless, these are both examples of consumer–resource interactions; most competition, by the way, is thought to operate through uptake of shared, limiting resources [120,200].

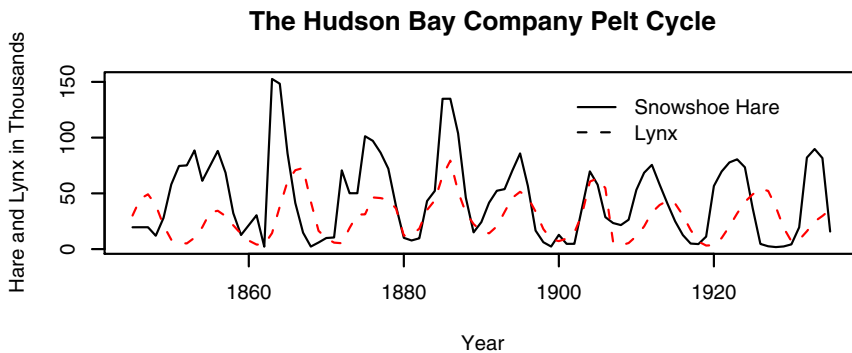


Fig. 6.1: Lynx–snowshoe hare cycles.

One of the most famous examples of species interactions in all of ecology is the lynx–snowshoe hare cycle, based on data from the Hudson Bay Trading

Co. trapping records (Fig. 6.1).¹ For decades, the lynx–hare cycle was used as a possible example of a predator–prey interaction, until a lot of hard work by a lot of people [94, 191] showed an asymmetric dynamic — while the lynx depends quite heavily on the hare, and seems to track hare abundance, the hare cycles seem to be caused by more than just lynx.

In this chapter, we will do a few things. First, we will cover various flavors of cosumer–resource models of different levels of complexity, and style. Also known as enemy–victim relations, or exploitative interactions, we represent cases in which one species has a negative effect and one a positive effect on the other. In doing so, we will illustrate some fundamental concepts about consumer–resource dynamics, such as how predators respond to prey abundances in both a *numerical* and a *functional* manner. We will try to develop an understanding of the range of dynamics for both continuous and discrete time systems.

6.1 Predators and Prey

This section covers a brief introduction to the classic predator–prey models, the Lotka–Volterra model, and the Rosenzweig–MacArthur extension.

6.1.1 Lotka–Volterra model

The Lotka–Volterra predator–prey model [117] is the simplest consumer–resource model we will cover. It is useful for a variety of reasons. First, its simplicity makes it relatively easy to explain. Second, it lays the groundwork for other consumer–resource interactions. Third, and perhaps more importantly, it captures a potentially fundamental feature of these interactions — instability. When prey reproduce and are limited only by the predator, and the predators are limited only by the abundance of prey, these interactions are not stable. This is, one could argue, the fundamental component of a predator–prey interaction. Only when you add additional, albeit realistic, factors (which we will cover later) do you get stable consumer–resource interactions. It seems to be true that reality helps stabilize the interactions, but at the core of the interaction is a tension of instability.

The Lotka–Volterra predator–prey model is relatively simple.

$$\frac{dH}{dt} = bH - aPH \quad (6.1)$$

$$\frac{dP}{dt} = eaPH - sP \quad (6.2)$$

Here the prey is H (herbivore, perhaps), the predator is P , and b , a , e , and s are parameters that we will discuss shortly. Let’s break these equations down. First, we examine the model with regard to its terms that increase growth (+

¹ These data are actually collected from a number of different regions and embedded with a complex food web, so it probably doesn’t make much sense to think of this as only one predator–prey pair of populations.

terms) and those that reduce growth rate ($-$ terms). Second, we find the terms where death of prey (in the prey equation) relates to growth of predators (in the predator equation).

The prey equation (eq. 6.1) has only two terms, bH and aPH . Therefore the prey population is growing at an instantaneous rate of bH . What does this mean? It means that its per capita rate is density independent, that is, it is growing exponentially, in the absence of the predator. This, as you know, frequently makes sense over very short time periods, but not over the long term. The units of b are similar to those for r — number of herbivores produced per herbivore.

What about the loss term, aPH ? This term is known as *mass action*, a term borrowed from chemistry, where the rate of a reaction of two substances (e.g., P and H) is merely a linear function, a , of their masses. That is, given the abundances of each population, they encounter each other *with the outcome of prey death* at instantaneous rate a . Thus a is frequently known as the kill rate or “attack” rate². The units of a are number of herbivores killed per predator per herbivore. When multiplied by PH , the units of the term become number of herbivores.

Lotka–Volterra predator–prey model

Here we create a function for the Lotka–Volterra predator–prey model that we use in `ode`.

```
> predpreyLV <- function(t, y, params) {
+   H <- y[1]
+   P <- y[2]
+   with(as.list(params), {
+     dH.dt <- b * H - a * P * H
+     dP.dt <- e * a * P * H - s * P
+     return(list(c(dH.dt, dP.dt)))
+   })
+ }
```

See the Appendix, B.10 for more information about using `ode`.

Functional response

What we have just described is the functional response of the predator [77]. The functional response is the rate at which a single predator kills prey. We can represent this a graph of the relation between the number of prey killed per predator per unit time (y -axis) vs. the prey available (x -axis) — the predators kill prey at rate aPH , so a single predator kills prey at rate aH . This simple relation is linear, and is known as a *type I* functional response (Fig. 6.2a). What does this assume about predator behavior? For one, it assumes that no matter how abundant prey become, a single predator can and will always kill

² In more detailed models, we may distinguish between encounter rate, attack rate, kill rate, and handling time, adding greatly to the detail of the model.

a fixed proportion, whether there are 0.1 or 100 prey· m^2 available. Other types of functional responses are often discussed, including types II and III; these saturate (reach an asymptote) at high prey densities, and we will discuss these below.

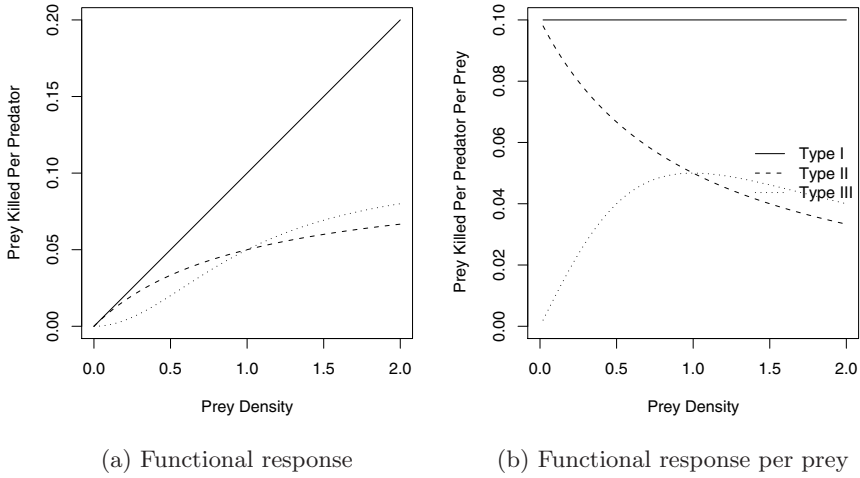


Fig. 6.2: Types I, II, and III predator functional responses; these are the rates at which predators kill prey across different prey densities. (a) The original functional responses; (b) Functional responses on a per prey basis. The Lotka–Volterra model assumes a type I functional response.

Ofttimes, especially with messy data, it is very difficult to distinguish among functional response types, especially at low prey densities [87]. It is sometimes easier to distinguish among them if we examine the functional responses *on a per prey basis*. This results in a much more stark contrast among the functional responses (Fig. 6.2b), at low prey densities.

Functional responses

This will graph types I, II, and III predator functional responses. Parameter a is the attack rate for mass action (type I). Parameter w is the maximum rate and D is the half saturation constant for the Michaelis-Menten/Holling disc equation in types II and III.

```
> a <- 0.1
> w <- 0.1
> D <- w/a
> curve(a * x, 0, 2, xlab = "Prey Density",
+       ylab = "Prey Killed Per Predator")
> curve(w * x/(D + x), 0, 2, add = TRUE, lty = 2)
> curve(w * x^2/(D^2 + x^2), 0, 2, add = TRUE, lty = 3)
```

It is sometimes easier to distinguish among these if we examine the per attack rate per prey, as a function of prey density.

```
> curve(w * x^2/(D^2 + x^2)/x, 0, 2, ylim = c(0, a), lty = 3,
+       xlab = "Prey Density", ylab = "Prey Killed Per Predator Per Prey")
> curve(w * x/(D + x)/x, 0, 2, lty = 2, add = TRUE)
> curve(a * x/x, 0, 2, add = TRUE, lty = 1)
> legend("right", c("Type I", "Type II", "Type III"), lty = 1:3,
+       bty = "n")
```

Numerical response

The *numerical response* of the predators is merely the population-level response of predators to the prey, that derives from both the growth and death terms.

In the Lotka–Volterra model, predators attack, kill or capture prey at rate a , but for the population to grow, they need to convert that prey into new predator biomass or offspring. They assimilate the nutrients in prey and convert the prey into predator body mass or offspring at rate e . Thus e is the efficiency (assimilation or conversion efficiency) of converting dead prey into new predator mass or predator numbers. The units of e are derived easily. Recall from above the units of a . The units of e must take the expression aPH with its units of numbers of herbivores, and give $eaPH$ units of numbers of predators. Therefore the units of e are numbers of predators per number of herbivores killed.

In this model, we pretend that predators die at a constant proportional rate, that is, at a constant per capita rate, s , in the absence of prey. At first blush, this may seem quite unrealistic. If, however, we consider that (i) not all predators are the same age, size, body condition, or occupying the same habitat, and (ii) they are all different genotypes, then we might expect many to die quickly, and a few to hang on for much longer. The exponential decay described by this simple formula of constant proportional loss may be a very good place to start when considering loss from a complex system such as a population. The numerical response is the combined response of this loss and the growth term.

Lotka–Volterra isoclines

What does this model tell us about the forces that govern the dynamics of predator–prey populations? Are there attractors or repellers? What can we learn?

As we saw in previous chapters, a good place to start learning about species interactions is with the ZNGIs or zero net growth isoclines. These tell us when populations tend to decrease or increase. How do we determine these ZNGIs for predator–prey interactions? We do so in the same fashion as for single species models, and for Lotka–Volterra competition — by setting the growth equations equal to zero and solving for the relevant state variable. Here we set the “herbivore” population growth rate equal to zero and solve for H^* — this is the value of H when $\dot{H} = 0$. As with L–V competition, it may be an isocline, a line along which all points in P and H space (the combinations of P and H) $\dot{H} = 0$. Solving for H ,

$$0 = bH - aPH \quad (6.3)$$

$$0 = H(b - aP) \quad (6.4)$$

$$H = 0 \quad (6.5)$$

It seems at first that the only solution is for extinction of herbivores. What other condition, however, allows $\dot{H} = 0$? Further rearrangement shows us that this will hold if $(b - aP) = 0$, that is,

$$P = \frac{b}{a} \quad (6.6)$$

then this provides the isocline for which $\dot{H} = 0$ (Fig. 6.3). Note that it depends only on the traits of the predator and prey, or the ratio of the intrinsic growth rate, b , and the attack rate, a . The faster the prey grows, the more predators you need to keep them in check. The more vulnerable the prey are to predators, or the better the hunters are, the fewer predators are needed to keep the prey in check.

Now let’s solve for the predator isocline, the set of points at which predator growth rate stops. We set $\dot{P} = 0$ and solve

$$\begin{aligned} 0 &= eaPH - sP \\ 0 &= P(eaH - s) \\ P = 0, \quad H &= \frac{s}{ea} \end{aligned} \quad (6.7)$$

What conditions cause the predator growth rate to equal zero? One equilibrium occurs at $P^* = 0$, and the other occurs at $H = s/(ea)$. As we saw for the herbivore, the state at which $\dot{P} = 0$ is independent of the predator population itself, but rather depends upon species’ traits only. It depends upon the density-independent per capita death rate s , conversion efficiency, e , and the attack rate, a . The higher the predator’s death rate, s , the more herbivores are required to maintain predator growth. The more efficient the predator is at assimilating

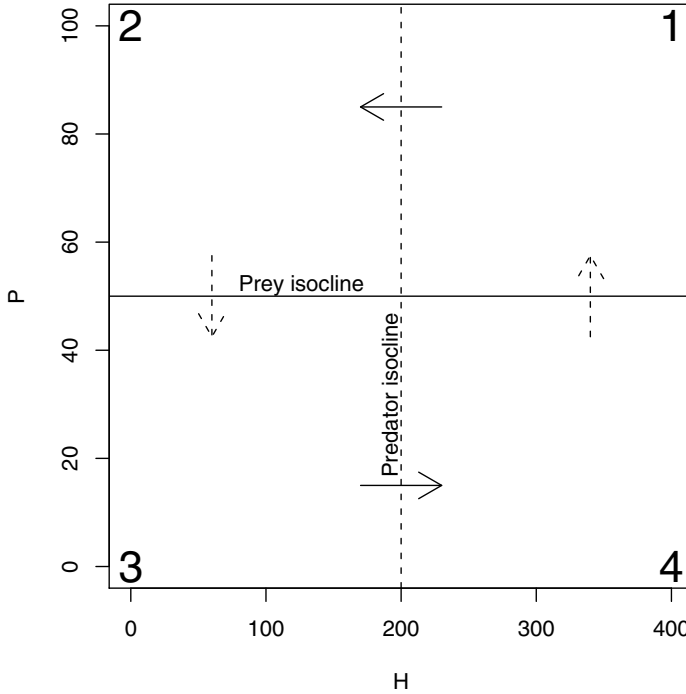


Fig. 6.3: Lotka–Volterra predator–prey isoclines. The isoclines (solid and dashed lines) are the set of all points for which the predator growth rate or the herbivore growth rate are zero. Increases and decreases in abundance are indicated by arrows (solid - prey, dashed - predator). Prey abundance, H , decreases in quadrants 1 and 2 because predator abundance, P , is high; prey abundance increases in quadrants 3 and 4 because predator abundance is low. In contrast, predator abundance, P , increases in quadrants 4 and 1 because prey abundance is high, whereas predator abundance decreases in quadrants 2 and 3 because prey abundance is low. These reciprocal changes in each species abundance results in counterclockwise dynamics between the two populations.

prey and converting them to biomass or progeny, e , the fewer prey are needed to maintain predator growth. The more efficient they are at encountering, attacking and killing prey, a , the fewer prey are required to maintain predator growth. We could also flip that around to focus on the prey, and state that the more nutritious the prey, or the more vulnerable they are to attack, the fewer are needed to maintain the predator population.

Fig. 6.3 illustrates the concepts we described above — predators increase when they have lots of food, and die back when food is scarce.

Lotka–Volterra prey and predator isoclines (Fig. 6.3)

We first select parameters and calculate these isoclines.

```
> b <- 0.5; a <- 0.01
> (Hi <- b/a)
```

```
[1] 50
```

```
> e <- 0.1; s <- 0.2
> (Pi <- s/(e * a))
```

```
[1] 200
```

We then set up an empty plot with plenty of room; we next add the prey isocline and arrows indicating trajectories. Last, we add the predator isocline, text, arrows, and then label each quadrant.

```
> plot(c(0, 2 * Pi), c(0, 2 * Hi), type = "n", xlab = "H",
+      ylab = "P")
> abline(h = Hi)
> text(Pi, Hi, "Prey isocline", adj = c(1.3, -0.3))
> arrows(x0 = c(0.85 * Pi, 1.15 * Pi), y0 = c(0.3 * Hi, 1.7 *
+      Hi), x1 = c(1.15 * Pi, 0.85 * Pi), y1 = c(0.3 * Hi, 1.7 *
+      Hi), len = 0.2)
> abline(v = Pi, lty = 2)
> text(Pi, Hi, "Predator isocline", adj = c(1.1, -0.2), srt = 90)
> arrows(x0 = c(0.3 * Pi, 1.7 * Pi), y0 = c(1.15 * Hi, 0.85 *
+      Hi), x1 = c(0.3 * Pi, 1.7 * Pi), y1 = c(0.85 * Hi, 1.15 *
+      Hi), lty = 2, len = 0.2)
> text(x = c(2 * Pi, 0, 0, 2 * Pi), y = c(2 * Hi, 2 * Hi, 0,
+      0), 1:4, cex = 2)
```

What do the dynamics in Fig. 6.3 tell us? Follow the path of the arrows. First, note that they cycle — they go around and around in a counterclockwise fashion, as each population responds to the changing abundances of the other species. We don't know much more than that yet, but we will later. The counterclockwise direction illustrates a negative feedback loop between predators and prey.

Next we use linear stability analysis to learn more about the long-term behavior of this interaction. We will use this analysis to compare several different predator–prey models.

6.1.2 Stability analysis for Lotka–Volterra

In this section, we will perform the necessary analytical work to understand the dynamics of Lotka–Volterra predator–prey dynamics, and we follow this up a peek at the time series dynamics to confirm our understanding based on the analytical work.

As before (e.g., Chapter 5), we can follow four steps: determine equilibria, create the Jacobian matrix, and solve and use the Jacobian.

Lotka–Volterra equilibrium

As you recall from Chapter 5, all we have to do is to solve the isoclines for where they cross. Thus we could set these equations equal to each other. It turned out, however, that the isoclines were so simple that we find that the prey and predator will come to rest at the (x, y) coordinates, $(b/a, s/(ea))$.

Creating, solving and using the Jacobian matrix

Take a look at the growth equations again (eqs. 6.1, 6.2). Here we take the partial derivatives of these because we want to know how each population growth rate changes in response to changes in the abundance each of the other population. The partial derivatives of the herbivore growth equation, with respect to itself and to the predator, go into the first row of the matrix, whereas the partial derivatives of the predator growth rate, with respect to the herbivore and itself go into the second row.³

$$\begin{pmatrix} \frac{\partial \dot{H}}{\partial H} & \frac{\partial \dot{H}}{\partial P} \\ \frac{\partial \dot{P}}{\partial H} & \frac{\partial \dot{P}}{\partial P} \end{pmatrix} = \begin{pmatrix} b - aP & -aH \\ eaP & eaH - s \end{pmatrix} \quad (6.8)$$

We can replace the P and H in the Jacobian with the equilibria found above. When we do this, we get

$$\begin{pmatrix} b - a(b/a) & -a(s/(ae)) \\ ea(b/a) & ea(s/(ae)) - s \end{pmatrix} = \begin{pmatrix} 0 & -s/e \\ eb & 0 \end{pmatrix}. \quad (6.9)$$

Typically a system will be more stable if the diagonal elements are more negative — that would mean that each population is self regulating, and it corresponds to the Routh-Hurwitz criterion,⁴

$$\mathbf{J}_{11} + \mathbf{J}_{22} < 0. \quad (6.10)$$

We notice that in eq. 6.9 these diagonal elements are both zero; these zeroes reveal that there is no negative density dependence within each population; that is no self-regulation.

The other part of the Routh-Hurwitz criteria is the condition,

$$\mathbf{J}_{11}\mathbf{J}_{22} - \mathbf{J}_{12}\mathbf{J}_{21} > 0. \quad (6.11)$$

In the predator–prey context, this suggests that the herbivore *declines* due to the predator ($\mathbf{J}_{12} < 0$) and the predator *increases* due to the herbivore ($\mathbf{J}_{21} > 0$). The signs of these elements make their product negative, and help make the

³ Recall that a partial derivative is just a derivative of a derivative, with respect to another state variable. In this case, it is not “the second derivative” *per se*, because that would be with respect to time, not with respect to one of the populations.

⁴ See Chapter 5 for an earlier use of the Routh-Hurwitz criteria

above condition true. Note that because $\mathbf{J}_{11}\mathbf{J}_{22}$, this condition reduces to $bs > 0$. Thus it seems that this will be true as long as both b and s are positive (which is always the case).

If we performed eigenanalysis on the above Jacobian matrix, we would find that the eigenvalues are complex (see next box). Because they are complex, this means that the populations will oscillate or cycle, with period $2\pi/\omega$ (Table 5.2). Because the real parts are zero, this means that the Lotka–Volterra predator–prey exhibits neutral stability (Table 5.2). Recall that neutral stability is the “in-between” case where perturbations at the equilibrium neither grow nor decline over time.

Lotka–Volterra predator–prey eigenanalysis

We can perform eigenanalysis given the parameters above.

```
> Jac <- matrix(c(0, -s/e, e * b, 0), byrow = TRUE, nr = 2)
> eigen(Jac)[["values"]]

[1] 0+0.3162i 0-0.3162i
```

Lotka–Volterra Dynamics

What do predator–prey cycles look like (Figs. 6.4a, 6.4b)? Typically, prey achieve higher abundances than predators — this makes sense if the “predators” are not pathogens (see Disease, below). It also makes sense when we assume that predators are not perfectly efficient at converting prey to offspring — that is, they have to metabolize a lot of energy per offspring ($e \ll 1$). Another characteristic we note is that the predator populations lag behind the prey — for instance, the prey peak occurs before the predator peak (Fig. 6.4a).

Lotka–Volterra predator–prey dynamics (Fig. 6.4a)

Here we set parameters and integrate the populations, with initial abundances of $H_0 = 25$, $P_0 = 5$.

```
> params1 <- c(b = b, a = a, s = s, e = e)
> Time <- seq(0, 100, by = 0.1)
> LV.out <- ode(c(H0 = 25, P0 = 5), Time, predpreyLV, params1)
```

Next we graph the populations over time.

```
> matplot(Time, (LV.out[, 2:3]), type = "l", ylab = "Population Size")
```

What do *neutral cycles* look like? Both populations oscillate indefinitely, going neither toward extinction, nor toward a stable *node* or point (Fig. 6.4b). An odd characteristic is that we could choose arbitrarily any new initial abundance, and the populations would continue to cycle on a new trajectory, passing through these new abundances every period (Fig. 6.4b).

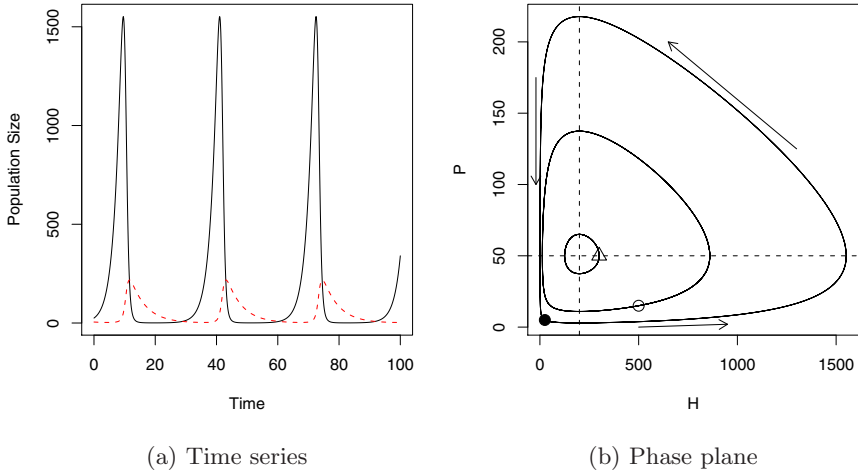


Fig. 6.4: Dynamics of the Lotka–Volterra predator–prey model. Both figures result from the same set of model parameters. (a) The times series shows the population sizes through time; these dynamics correspond to the largest oscillations in (b). (b) The phase plane plot includes three different starting abundances, indicated by symbols; the largest cycle (through solid dot) (a).

Lotka–Volterra predator–prey dynamics (Fig. 6.4b)

We integrate the same model as above twice more, but with arbitrarily different starting abundances — everything else is the same.

```
> LV.out2 <- ode(c(H0 = 500, P0 = 15), Time, predpreyLV, params1)
> LV.out3 <- ode(c(H0 = 300, P0 = 50), Time, predpreyLV, params1)
```

Now we plot the *phase plane portrait* of the first predator–prey pair, add trajectories associated with different starting points, and finally add the isoclines.

```
> plot(LV.out[, 2], LV.out[, 3], type = "l", ylab = "P", xlab = "H")
> points(25, 5, cex = 1.5, pch = 19)
> arrows(x0 = c(1300, -20, 500), y0 = c(125, 175, 0), x1 = c(650,
+   -20, 950), y1 = c(200, 100, 2), length = 0.1)
> lines(LV.out2[, 2], LV.out2[, 3])
> points(500, 15, cex = 1.5)
> lines(LV.out3[, 2], LV.out3[, 3])
> points(300, 50, cex = 1.5, pch = 2)
> abline(h = b/a, lty = 2)
> abline(v = s/(e * a), lty = 2)
```

6.1.3 Rosenzweig–MacArthur model

Michael Rosenzweig, a graduate student at the time, and his adviser, Robert MacArthur, proposed a predator–prey model that added two components to the

Lotka–Volterra model [177, 180]. First, they felt that in the absence of predators, prey would become self-limiting.

A second element added to the Lotka–Volterra model was a saturating functional response of the predators to prey density (Fig. 6.2). They felt that if prey density became high enough, predators would reach a maximum number of prey killed per unit time. First, predator appetite could become satiated, and second, many predators need time to *handle* prey (catch, subdue, and consume it). The time required to do this is referred to as *handling time* and may not change with prey density. Therefore the predator would ultimately be limited not by prey density, but by its own handling time. Foraging theory and species-specific predator–prey models explore these components of predator behavior [72, 123]. These realities (for some predators) cause an upper limit to the number of prey a predator can consume per unit time. This limitation is known as a type II functional response [77] (Fig. 6.2), and may take a variety of different forms — any monotonically saturating function is referred to as a type II functional response.

Here we add prey self-limitation by using logistic growth. We add the type II functional response using the Michaelis-Menten or Monod function.⁵

$$\frac{dH}{dt} = bH(1 - \alpha H) - w \frac{H}{D + H} P \quad (6.12)$$

$$\frac{dP}{dt} = ew \frac{H}{D + H} P - sP \quad (6.13)$$

$$(6.14)$$

where w and D are new constants.

Here we focus on the functional response, $w \frac{H}{D+H}$ (Fig. 6.2). How can we interpret this fraction? First let's consider what happens when prey density is very high. We do that by examining the limit of the functional response as prey density goes to infinity. As H gets really big, $D + H$ gets closer to H because D stays relatively small. In contrast, the numerator wH will continue to grow as a multiple of w , so w will always be important. Therefore, the functional response reduces to $wH/H = w$, or we state

$$\lim_{H \rightarrow \infty} w \frac{H}{D + H} = \frac{wH}{H} = w. \quad (6.15)$$

That is a pretty convenient interpretation, that w is the maximum capture rate.

What is the shape of the function at low prey abundance, or as $H \rightarrow 0$? H becomes smaller and smaller, and D becomes relatively more and more important. Therefore, the functional response reduces to $(w/D)H$, or

$$\lim_{H \rightarrow 0} w \frac{H}{D + H} = \frac{w}{D} H \quad (6.16)$$

Note that this is a linear functional response where $w/D = a$, where a was the attack rate in the Lotka–Volterra type I functional response aH . Cool!

⁵ An alternative parameterization is the Holling disc equation $aHP/(1 + bH)$.

Rosenzweig-MacArthur predator-prey function

Before we move on, let's make an R function of the Rosenzweig-MacArthur model.

```
> predpreyRM <- function(t, y, p) {
+   H <- y[1]
+   P <- y[2]
+   with(as.list(p), {
+     dH.dt <- b * H * (1 - alpha * H) - w * P * H / (D +
+     H)
+     dP.dt <- e * w * P * H / (D + H) - s * P
+     return(list(c(dH.dt, dP.dt)))
+   })
+ }
```

Rosenzweig-MacArthur isoclines

As usual, our first step is to find the zero net growth isoclines — we will first derive them, and then discuss them. We can begin with the prey⁶, setting $\dot{H} = 0$.

$$\begin{aligned}
 0 &= bH(1 - \alpha H) - w \frac{PH}{D + H} \\
 P &= \frac{(D + H)}{w} b(1 - \alpha H) \\
 P &= \frac{b}{w} (D + (1 - \alpha D)H - \alpha H^2)
 \end{aligned} \tag{6.17}$$

You will notice that it is a quadratic equation, and we can see that it is concave down ($-\alpha H^2$, when $\alpha > 0$), and the peak typically occurs at some $H > 0$.⁷

Next we find the predator isocline.

$$\begin{aligned}
 0 &= ew \frac{PH}{D + H} - sP \\
 0 &= P \left(\frac{ewH}{D + H} - s \right)
 \end{aligned}$$

One equilibrium is $P = 0$; let's focus on the other, where $\frac{ewH}{D+H} - s = 0$. This is telling us that the population growth rate of the predator will be zero when...? Right — when H takes on a value such that this expression is true. To find that value, we solve this part of the expression for H .

$$\begin{aligned}
 0 &= \frac{ewH}{D + H} - s \\
 ewH - sH &= sD \\
 H &= \frac{sD}{ew - s}
 \end{aligned} \tag{6.18}$$

Thus $\dot{P} = 0$ when $H = sD/(ew - s)$. It is thus the straight line and independent of P — it depends entirely on the parameters.

⁶ or as Rosenzweig liked to say, the “victim”

⁷ Recall the rules you learned long ago, regarding the quadratic equation....

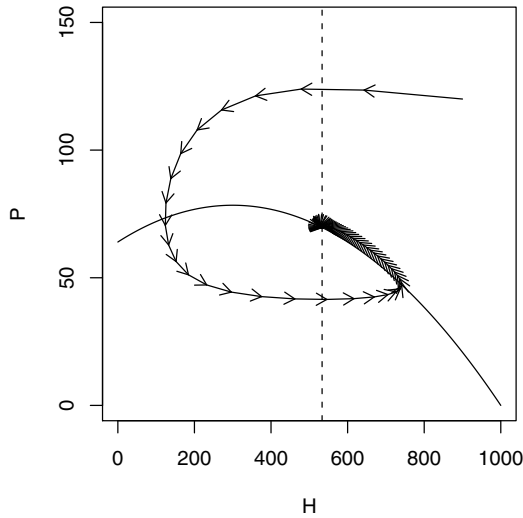


Fig. 6.5: Rosenzweig-MacArthur predator–prey isoclines for the predator (dashed) and the prey (solid). The isoclines are the set of all points for which the predator growth rate (dashed) and the herbivore growth rate (solid) are zero. The arrows get shorter, and the arrowheads closer together because the populations change more slowly as we approach the steady state equilibrium. Note that the x -intercept of the prey isocline is K .

Consider the dynamics around these isoclines (Fig. 6.5). Whenever the prey are abundant (left of dashed predator isocline), the predators increase, and when the prey are rare, the predators decrease. In contrast, whenever the predators are abundant (above solid prey isocline), then prey decline, and when predators are rare, then prey increase. In this case (Fig. 6.5), we see the classic counter-clockwise cycling through state space leading to a stable point equilibrium.

Rosenzweig-MacArthur isoclines (Fig. 6.5)

Let's graph the zero net growth isoclines for this model. First we set parameters, and make an expression for the prey isocline.

```
> b <- 0.8; e <- 0.07; s <- 0.2; w <- 5;
> D <- 400; alpha <- 0.001; H <- 0:(1/alpha)
> Hiso <- expression(b/w * (D + (1 - alpha * D) * H - alpha *
+   H^2))
> HisoStable <- eval(Hiso)
```

We also want to add a single trajectory, so we integrate using `ode`.

```
> p.RM <- c(b = b, alpha = alpha, e = e, s = s, w = w, D = D)
> Time <- 150
> RM1 <- ode(c(900, 120), 1:Time, predpreyRM, p.RM)
```

Finally, we plot everything, starting with the isoclines, and adding the trajectory using `arrows`.

```
> plot(H, HisoStable, type = "l", ylab = "P", xlab = "H", ylim = c(0,
+   150))
> abline(v = s * D/(e * w - s), lty = 2)
> arrows(RM1[-Time, 2], RM1[-Time, 3], RM1[-1, 2], RM1[-1,
+   3], length = 0.1)
```

Note that an arrow illustrates the change per one unit of time because we chose to have `ode` return H, P at every integer time step.

Creating and using the Jacobian

As we did for the Lotka–Volterra system, here we demonstrate stability by analyzing the Jacobian matrix of partial derivatives of each populations growth rate with respect to each other. We put all these together in a matrix, and it looks like this.

$$\begin{pmatrix} \frac{\partial \dot{H}}{\partial H} & \frac{\partial \dot{H}}{\partial P} \\ \frac{\partial \dot{P}}{\partial H} & \frac{\partial \dot{P}}{\partial P} \end{pmatrix} = \begin{pmatrix} b - 2\alpha b H - \left(\frac{wP}{D+H} - \frac{wPH}{(D+H)^2} \right) & -w \frac{H}{D+H} \\ \frac{ewP}{D+H} - \frac{ewPH}{(D+H)^2} & ew \frac{H}{D+H} - s \end{pmatrix} \quad (6.19)$$

You could review your rules of calculus, and prove to yourself that these *are* the correct partial derivatives.

Analysis of the Jacobian for Rosenzweig-MacArthur

Here we are going to write *expressions* for the to time derivatives, make a *list* of all the partial derivatives, find the stable equilibrium point, evaluate the partial derivatives as we stick them into the Jacobian matrix, and then perform eigenanalysis on the Jacobian. First, the time derivatives.

```
> dhdt <- expression(b * H * (1 - alpha * H) - w * P * H/(D +
+ H))
> dpdt <- expression(e * w * P * H/(D + H) - s * P)
```

Next we create a *list* of the partial derivatives, where their order will allow us, below, to fill columns of the Jacobian matrix.

```
> RMjac1 <- list(D(dhdt, "H"), D(dpdt, "H"), D(dhdt, "P"),
+ D(dpdt, "P"))
```

We need the stable equilibria, H^* , P^* . We know the value of H^* right away, because the predator isocline is a constant. Predator abundance exhibits zero growth when $H = sD/(ew - s)$, or

```
> H <- s * D/(e * w - s)
```

Know all we have to do is substitute that into the herbivore isocline, and we are finished.

```
> P <- eval(Hiso)
```

Now we “apply” the `eval` function to each *component* of the *list* of PD’s, using the values of H and P we just created. We stick these values into a matrix and perform eigenanalysis.

```
> (RM.jac2 <- matrix(sapply(RMjac1, function(pd) eval(pd)),
+ nrow = 2))

      [,1] [,2]
[1,] -0.2133 -2.857
[2,]  0.0112  0.000

> eigen(RM.jac2)[["values"]]

[1] -0.1067+0.1436i -0.1067-0.1436i
```

If we substitute in all of the relevant parameters and equilibria (see previous box), we would find that at the equilibrium, the Jacobian evaluates to

```
> RM.jac2

      [,1] [,2]
[1,] -0.2133 -2.857
[2,]  0.0112  0.000
```

where we note⁸ that the effect of both the prey itself and predators on the prey population growth rate (at the equilibrium) are negative, whereas the effect

⁸ Recall that species 1 is in column 1 and row 1, and we interpret this as the effect of [column] on [row]

of prey on predators is positive, and the predators have no effect at all on themselves. How does this compare to the Jacobian matrix for Lotka–Volterra? There, the effect of prey on itself was zero.

If we examine the eigenvalues of this matrix,

```
> eigen(RM.jac2)[["values"]]
[1] -0.1067+0.1436i -0.1067-0.1436i
```

We find that for *these parameter values* the dominant eigenvalue is negative, indicating that the equilibrium is a stable attractor (Table 5.2). The presence of the imaginary part shows a cyclical approach toward the equilibrium — we could calculate the period of that cycle, if we wanted to (Table 5.2).

6.1.4 The paradox of enrichment

Rosenzweig and MacArthur [177, 180] realized that the type II functional response of the predator actually could *destablize* the interaction. They focused primarily on traits (or parameters) that bring the predator isocline closer to the y-axis, including predator assimilation efficiency —“overly” efficient predators (e.g., large e) can drive prey extinct.

In 1971, Rosenzweig described the “paradox of enrichment” where simply increasing the carrying capacity of the prey could destabilize the system [179]. In this case, the saturating nature of the functional response allows the prey to escape control when all individual predators become saturated — this allows prey to achieve higher peak abundances, until they reach carrying capacity. At that point, the predators can eventually achieve high abundance and drive the prey back down. In addition, when predators have a low half-saturation constant (small D) this allows them to kill a higher proportion of the prey when the prey are rare, thus driving the prey to even lower levels.

When would predator–prey cycles be stable, and when would they explode and destabilize? It all has to do with the position of the predator isocline relative to the hump of the prey isocline. When the hump or peak of the prey isocline is inside the predator isocline, the system has a stable point attractor (Fig. 6.5). When the carrying capacity of the prey increases, it draws the hump to the outside the predator isocline (Fig. 6.6a). This eliminates the stable point attractor (Fig. 6.6b), the cycles get larger, nevering return to the equilibrium.

If we focus primarily on “enrichment” of the prey population, that means increasing carrying capacity or decreasing the self-limitation (decreasing α). If we look at the Jacobian matrix (eq. 6.30), we notice that α occurs only in the prey’s negative effect on itself. It is common sense that this negative feedback of the prey upon itself is an important mechanism enhancing stability⁹, preventing large oscillations. What we see in the Jacobian matrix is that *decreasing α reduces self regulation in the system*.

The phase plane portrait (Fig. 6.6a) reveals that the equilibrium, where the isoclines cross, is a repeller, an unstable equilibrium — the populations spiral away from the equilibrium.

⁹ Recall the Routh-Hurwitz criteria, $\mathbf{J}_{11} + \mathbf{J}_{22} < 0$, $\mathbf{J}_{11}\mathbf{J}_{22} - \mathbf{J}_{12}\mathbf{J}_{21} > 0$

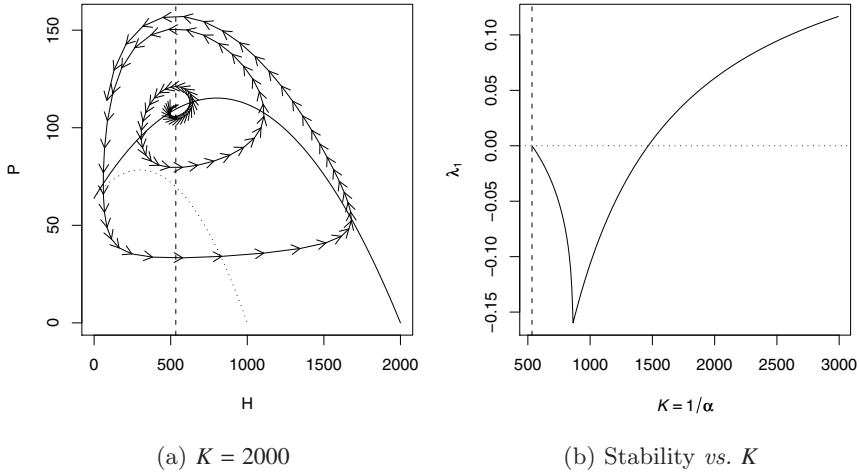


Fig. 6.6: Illustrating the paradox of enrichment. (a) One example of the paradox of enrichment, where large carrying capacity causes cycles instead of a stable attractor (compare to Fig. 6.5). (b) Stability declines when the prey are too strongly self-limiting (very small K) or especially when they have the capacity to achieve very high abundances (large K).

Phase plane portrait for the paradox of enrichment (Fig. 6.6a)

Let's graph the zero net growth isoclines for this model. We use previous parameter values, but change α , and reevaluate range of H we need, and the new trajectory.

```
> p.RM["alpha"] <- alpha <- 5e-04
> Time <- 100
> H <- 0:(1/alpha)
> RM2 <- ode(c(500, 110), 1:Time, predpreyRM, p.RM)
```

Next, we plot everything, starting with the prey isoclines with large K (solid line) and then the small K (dotted line). Last, we add the trajectory for the system with large K , small α .

```
> plot(H, eval(Hiso), type = "l", ylab = "P", xlab = "H", ylim = c(0,
+   max(RM2[, 3])))
> lines(0:1000, HisoStable, lty = 3)
> abline(v = s * D/(e * w - s), lty = 2)
> arrows(RM2[-Time, 2], RM2[-Time, 3], RM2[-1, 2], RM2[-1,
+   3], length = 0.1)
```

The Jacobian for the paradox of enrichment (Fig. 6.6b)

Using the expressions created above, we vary α and examine the effect on λ_1 , the dominant eigenvalue. We select α s so that K ranges very small ($K = H^*$) to very large.

```
> H <- with(as.list(p.RM), s * D/(e * w - s))
> alphas <- 1/seq(H, 3000, by = 10)
```

For each α_i in our sequence, we calculate P^* , then evaluate the Jacobian, arranging it in a matrix. The result is a *list* of evaluated Jacobian matrices.

```
> RM.jacList <- lapply(1:length(alphas), function(i) {
+   alpha <- alphas[i]
+   P <- eval(Hiso)
+   matrix(sapply(RMjac1, function(pd) eval(pd)), nrow = 2)
+ })
```

For each evaluated Jacobian matrix, we perform eigenanalysis, and retain the maximum real part of the eigen values.

```
> L1 <- sapply(RM.jacList, function(J) max(Re(eigen(J)[["values"]])))
```

Finally, we plot these dominant eigenvalues *vs.* the carrying capacities that generated them.

```
> plot(1/alphas, L1, type = "l", xlab = quote(italic(K) ==
+   1/alpha), ylab = quote(lambda[1]))
> abline(h = 0, lty = 3)
> abline(v = H, lty = 2)
```

6.2 Space, Hosts, and Parasitoids

A parasitoid is a ghastly thing. These animals, frequently small wasps, characteristically lay an egg on their hosts, often a caterpillar, the young hatch out, and then slowly consume the juicy bits of the host. Neither the adult wasp nor their larvae immediately kill their host. Many eggs can be laid on a host, and many larvae can live (at least briefly) in the host. Ultimately, however, the host is gradually consumed and one or a few larvae per host metamorphosizes into an adult. Thus parasitoid larvae always consume and kill their hapless host. In this sense, their natural history is intermediate between that of predator and parasite — they are parasite-like, or a parasitoid. Thank the gods for parasitoids, as they often help suppress other animals that we like even less, such as herbivores that eat our crops.

There are a variety of characteristics about host–parasitoid relations that might make them different from the kind of predator–prey relations that we have been thinking about up until now. In particular, the life cycles of host and prey are so intimately connected that there is a one-to-one correspondence between dead hosts and the number of parasitoids in the next generation. If

we know how many hosts are killed, then we know approximately how many parasitoids are born.

6.2.1 Independent and random attacks

In keeping with convention of parasitoid models, let us describe dynamics of hosts and their parasitoid enemies with a discrete time model [131]. This makes sense, as these organisms, typically insects, frequently complete their life cycles together and within a year. Let us pretend that the hosts, H , grow geometrically in the absence of parasitoids, such that $H_{t+1} = RH_t$. If we assume that some individuals are attacked and killed by parasitoids, H_a , then this rate becomes

$$H_{t+1} = R(H_t - H_a) \quad (6.20)$$

Further, we assume a couple things about parasitoid natural history. First, we pretend the parasitoids *search randomly and independently* for prey, and are able to search area, a (the “area of discovery”), in their life time. The total number of *attack events* then is $E_t = aH_tP_t$. “Attack events” is sort of a strange way to put it but it makes perfect sense, given the natural history. Adults may lay an egg on any host it encounters, but this event does not kill the host. Therefore, hosts can receive more than one egg, thereby making the number of attacked hosts lower than the number of attack events, E_t . Each attack occurs as a random event, and for now we assume that each attack is independent of all others.

For many species of parasitoids, only one adult can emerge from a single host, *regardless of how many eggs were laid on that host*. Therefore, the number of emerging adult parasitoids, P_{t+1} , is simply the number of hosts that were attacked at all, whether the host received one egg or many. Therefore, the number of parasitoids at time $t + 1$ is merely the number of hosts that were attacked, and we can represent this as

$$P_{t+1} = H_a. \quad (6.21)$$

The assumption of one dead host = one new parasitoid can be easily relaxed if we introduce a constant q , such that $P_{t+1} = qH_a$.

Nicholson and Bailey [149] took advantage of a well known discrete probability distribution, the *Poisson distribution*,¹⁰ to create a simple discrete analytical model of host–parasitoid dynamics.

$$H_{t+1} = RH_t e^{-aP_t} \quad (6.22)$$

$$P_{t+1} = H_t (1 - e^{-aP_t}) \quad (6.23)$$

¹⁰ This distribution was discovered by a french mathematician (Siméon-Denis Poisson (1781–1840), so we pronounce the name “pwah-sohn,” like the “fois” of fois gras, and the “sohn” with a quiet “n,” like, well, like the French would say it. It is the series

$$\frac{1}{e^\mu}, \frac{\mu}{1!e^\mu}, \frac{\mu^2}{2!e^\mu}, \dots, \frac{\mu^r}{r!e^\mu}$$

representing the probabilities of occurrence of 0, 1, 2, . . . r .

Here aP_t is the mean number of attacks per larva (λ of the Poisson distribution) and results from P_t parasitoids searching a particular area, a . Parameter a is known as the “area of discovery,” the area searched during a parasitoid’s lifetime — more area searched (i.e., larger a) means more hosts killed, and fewer to survive and contribute to the next generation. The Poisson distribution tells us that if attacks are random and independent, then e^{-aP_t} is the probability of a host not being attacked. Therefore, $H_t e^{-aP_t}$ is the expected number of hosts which are not attacked and survive to reproduce.

We can find the equilibrium for this discrete model. Recall that, in models of continuous differential equations, this meant letting $dN/dt = 0$. In the discrete case, it means letting $N_{t+1} - N_t = 0$. The equilibrium, N^* , is the value of N when this is true, so $N^* = N_{t+1} = N_t$. We use this to find the equilibrium of the parasitoids, starting with eq. 6.22,

$$\begin{aligned} H^* &= RH^* e^{-aP_t} \\ 1 &= R e^{-aP_t} \\ P^* &= P_t = \frac{\log R}{a} \end{aligned} \tag{6.24}$$

If we recognize that in eq. 6.23, $P_{t+1} = H_t - H_{t+1}/R$, and that $H^* = H_t = H_{t+1}$, then we find the host equilibrium

$$H^* = P^* \frac{R}{R-1}. \tag{6.25}$$

The next section motivates the Nicholson-Bailey model with a simulation of these dynamics. It may provide a more intuitive understanding than simply throwing a probability distribution at the problem. In the end, we wind up at the same place, and this should be reassuring.

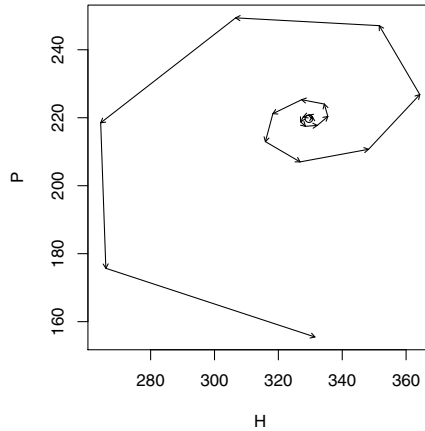


Fig. 6.7: Dynamics around the unstable equilibrium of the Nicholson-Bailey host–parasitoid model ($R = 3$, $a = 0.005$). Arrows indicate a single time step; the point is the equilibrium at the center.

Dynamics of the Nicholson-Bailey host–parasitoid model (Fig. 6.7)

This model assumes that parasitoids search a fixed area, and each attacks hosts randomly and independently. We set the duration of the time series, the model parameters, and create an empty data frame to hold the results.

```
> time <- 20
> R <- 3
> a <- 0.005
> HPs <- data.frame(Hs <- numeric(time), Ps <- numeric(time))
```

Next we calculate the equilibrium, and use a nearby point for the initial abundances.

```
> Pst <- log(R)/a
> Hst <- Pst * R/(R - 1)
> HPs[1, ] <- c(Hst + 1, Pst)
```

We then project the dynamics, one generation at a time, for each time step.

```
> for (t in 1:(time - 1)) HPs[t + 1, ] <- {
+   H <- R * HPs[t, 1] * exp(-a * HPs[t, 2])
+   P <- HPs[t, 1] * (1 - exp(-a * HPs[t, 2]))
+   c(H, P)
+ }
```

Last we plot the trajectory, placing a point at the (unstable) equilibrium, and using arrows to highlight the direction and magnitude of the increment of change per time step.

```
> plot(HPs[, 1], HPs[, 2], type = "n", xlab = "H", ylab = "P")
> points(Hst, Pst)
> arrows(HPs[-time, 1], HPs[-time, 2], HPs[-1, 1], HPs[-1,
+   2], length = 0.05)
```

Simulating Random Attacks

This section relies on code and could be skipped if necessary.

This model makes strong assumptions about the proportion of hosts that escape attacks, that depends on parasitoid attacks being random and independent. Therefore let us simulate parasitoid attacks that are random and independent and compare those data to field data. Let's start with some field data collected by Mark Hassell [131], which are the number of parasitoid larvae per host, either 0, 1, 2, 3, or 4 larvae.

```
> totals <- c(1066, 176, 48, 8, 5)
```

Here, 1066 hosts have no larvae. We can recreate the data set, where we have one observation per host: the number of larvae in that host.

```
> obs <- rep(0:4, totals)
```

To simulate random attacks, let's use the same number of hosts, and the same number of attacks, and let the computer attack hosts at random. We calculate the total number of hosts, H , and the total and mean number of attacks experienced by hosts.

```
> H <- sum(totals)
> No.attacks <- sum(obs)
> mean.attacks <- mean(obs)
```

Next, the predators “sample” the hosts at random and with equal probability. To code this, we identify the hosts by numbering them 1– H . We then attack these hosts independently, that is, we *replace* each attacked host back into the pool of possible prey.

```
> attack.events <- sample(x = 1:H, size = No.attacks, replace = TRUE)
```

We then count the number times different hosts were attacked.

```
> N.attacks.per.host <- table(attack.events)
```

and then find count the number of hosts that were attacked once, twice, or more.

```
> (tmp <- table(N.attacks.per.host))
```

```
N.attacks.per.host
 1  2  3
250 30  2
```

We see, for instance, that 30 hosts were attacked twice. This also allows us to know how many hosts were *not* attacked,

```
> not.att <- H - sum(tmp)
```

Let's compare the observed data to the simulated data.

```
> obs.sim <- rep(0:max(N.attacks.per.host), c(not.att, tmp))
> table(obs)
```

```
obs
  0    1    2    3    4
1066 176  48   8   5
```

```
> table(obs.sim)
```

```
obs.sim
  0    1    2    3
1021 250  30   2
```

There seem to be fewer unattacked hosts and more attacked hosts in the random-attack simulation than in the observed data. Is this simply a consequence of the observed values being one of many random possibilities? Or is it a systematic difference? How do we check this?

One way to check whether the observed data differ from our model of random and independent attacks is to simulate many such observations, and compare the observed data (1066 unattacked hosts) to the *distribution* of the simulated results. Here we do this by performing n simulations of the same steps we used above.

```
> n <- 1000
> unatt.sim <- sapply(1:n, function(j) {
+   host.sim.att <- sample(x = 1:H, size = No.attacks, replace = TRUE)
+   attacks.per <- table(host.sim.att)
+   n.attd <- length(attacks.per)
+   H - n.attd
+ })
```

Next we just make a histogram (Fig. 6.8) of the number of unattacked hosts, adding a dotted line to show where the true data lie, and a dashed line for the prediction, under the assumption of random and independent attacks, based on the Poisson distribution.

```
> hist(unatt.sim, xlab = "Simulated # of Unattacked Hosts",
+   prob = TRUE, xlim = c(1000, 1070))
> abline(v = 1066, lty = 3)
> abline(v = exp(-mean.attacks), lty = 2)
```

Our simulation results (Fig. 6.8) indicate that the observed data include far more unattacked hosts than expected by chance.

Another, quicker way to evaluate the assumption of independent and random attacks is to compare the ratio of the variance of the observed larvae per host to the mean. If the data follow a Poisson distribution, this ratio is expected to equal to one.

```
> (I <- var(obs)/mean(obs))
```

```
[1] 1.404
```

This ratio is greater than one, but we do not know if this could be due to chance. We can test it, because under the null hypothesis, we expect that the product of the ratio and the number of hosts, H , follows a χ^2 distribution, with $H - 1$ degrees of freedom. We can ask how likely this ratio, or a more extreme

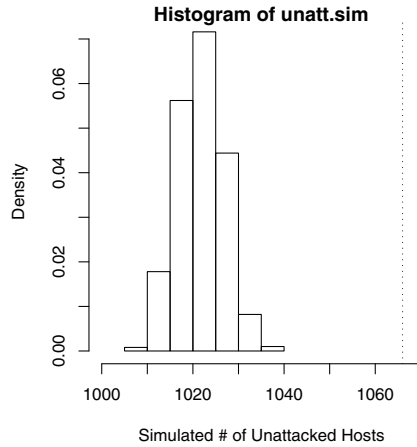


Fig. 6.8: Histogram of simulated host populations, attacked at a rate of 0.24 mean attacks per host, assuming an attacks on hosts are random and independent of each other.

ratio, would be, if attacks are random and independent. We compare it to the cumulative probability density function for the χ^2 distribution.

```
> 1 - pchisq(I * H, df = H - 1)
```

```
[1] 0
```

We find that, given this large number of observations (1303 hosts), it is exceedingly unlikely to observe a ratio this large or larger. It is nice that this agrees with our simulation! We feel more confident when a parametric test agrees with a simulation of our ideas; perhaps both are not necessary, but working through both helps us to understand what we are doing.

6.2.2 Aggregation leads to coexistence

The above model (eqs. 6.22, 6.23) has two problems. The first is that it doesn't reflect the biology — parasitoids tend to aggregate on particular hosts or on local populations of the hosts. Some hosts are more likely to get attacked than others, resulting in more unattacked hosts and more hosts receiving multiple attacks, than predicted under random and independent attacks. This may be related to their searching behavior, or to spatial distributions of hosts in the landscape. The second problem is that the above model is not stable, and predicts that the parasitoid or host becomes extinct. We know that in nature they don't become extinct! Perhaps we can kill two birds with one stone, and fix both problems with one step. That is what Robert May [131] and others have done, by assuming that parasitoids *aggregate*.

May proposed the following logic for one reason we might observe more unattacked hosts than expected. Imagine that the distribution of parasitoids

among patches in the landscape can be described with one probability distribution, with some particular mean and variance in the number of parasitoids per patch. Imagine next that their attacks on hosts *within* each patch are random and independent and thus described with the Poisson distribution. This will result in a compound distribution of attacks — a combination of the among-patch, and within-patch distributions. If we examine the distribution of larvae per host, for all hosts in the landscape, we will find a higher proportion of unattacked hosts, and a higher proportion of hosts that are attacked many times.

The *negative binomial distribution* can describe data, such as the number of larvae per host, in which *the variance is greater than the mean*. Thus the negative binomial distribution can describe greater aggregation than the Poisson distribution (where $\mu = \sigma^2$), and thereby describe nature somewhat more accurately. May suggested that while the true distribution of larvae in hosts, in any particular case, was unlikely to truly be a negative binomial, it was nonetheless a useful approximation.

Ecologists frequently use the negative binomial distribution for data where the variance is greater than the mean. There are different derivations of the distribution, and therefore different parameterizations [13]. In ecology, we typically use the mean, μ , and the *overdispersion* parameter k . The variance, σ^2 , is a function of these, where $\sigma^2 = \mu + \mu^2/k$; by overdispersion we mean that $\sigma^2 > \mu$. Large values of k ($k > 10$) indicate randomness, and the distribution becomes indistinguishable from the Poisson. Small values ($k < 2$) indicate aggregation.¹¹ Fig. 6.9 shows examples.

Showing the negative binomial distribution (Fig. 6.9)

R has the negative binomial distribution built in, but does not use k as one of its arguments; rather, `size = k`. Here we generate a graph showing the distribution with different values of k .

```
> nb.dat <- cbind(Random = dnbinom(0:5, mu = 0.5, size = 10^10),
+   `k=10` = dnbinom(0:5, mu = 0.5, size = 10), `k=1` = dnbinom(0:5,
+   mu = 0.5, size = 1), `k=0.01` = dnbinom(0:5, mu = 0.5,
+   size = 0.01))
> matplot(0:5, nb.dat, type = "b", pch = 1:4, col = 1, ylim = c(0,
+   1), xlab = "Attacks per Hosts", ylab = "Probability")
> legend("topright", rev(colnames(nb.dat)), pch = 4:1, lty = 4:1,
+   bty = "n")
> mtext(quote(mu == 0.5), padj = 2)
```

The proportion of unattacked hosts expected under the negative binomial is $(1 + aP_i/k)^{-k}$. Therefore, we can write the analytical expressions for the population dynamics that are very similar to those of the Nicholson-Bailey model, but using the negative binomial distribution.

¹¹ Although k is often referred to as a “clumping” or “aggregation” parameter, we might think of it as a *randomness* parameter, because larger values result in more random, Poisson-like distributions.

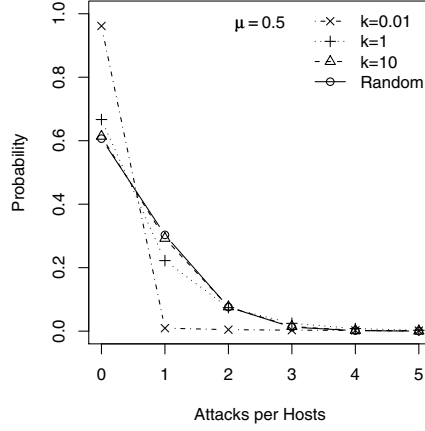


Fig. 6.9: The negative binomial distribution, where the dispersion parameter k controls variance or breadth the distribution. For a given mean, smaller k causes a greater variance, and results in a higher proportion of zeroes. These k are representative of values from Pacala *et al.* (1990).

$$H_{t+1} = RH_t \left(1 + \frac{aP_t}{k} \right)^{-k} \quad (6.26)$$

$$P_{t+1} = H_t - H_t \left(1 + \frac{aP_t}{k} \right)^{-k} \quad (6.27)$$

May [131] referred to this as a phenomenological model (as opposed to mechanistic) because the negative binomial merely approximates the true, perhaps compound, distribution.

Equilibria for a discrete-time model

The equilibria of the host and parasitoid populations in May's model (eqs. 6.26, 6.27) are derived simply, once we decide upon how to describe the condition for an equilibrium (a.k.a. steady state). In models of differential equations, this meant letting $dN/dt = 0$. In the discrete case it means letting $N_{t+1} - N_t = 0$. The equilibrium is the value of N when this is true, so $N^* = N_{t+1} = N_t$. Following this train of thought, we have $H^* = H_{t+1} = H_t$ and $P^* = P_{t+1} = P_t$.

To solve for equilibria, we begin with the expression for hosts (eq. 6.26), and solve for equilibrium parasitoid density. In this case, we can divide eq. 6.26 both sides by H^* .

$$\begin{aligned}
1 &= R \left(1 + \frac{aP_t}{k} \right)^{-k} \\
R^{-1} &= \left(1 + \frac{aP_t}{k} \right)^{-k} \\
R^{1/k} &= 1 + \frac{aP_t}{k} \\
P^* &= \frac{k}{a} (R^{1/k} - 1)
\end{aligned} \tag{6.28}$$

Given this, what causes increases in P^* ? Certainly decreasing a leads to increases in P^* . If a parasitoid population has a smaller a (i.e. smaller area of discovery), this means that they require less space, and can thereby achieve higher *density*. Increasing k means less aggregated attack events, which increases the probability that a parasitoid larvae is alone on a host, and this increases parasitoid density, but only up to a limit.

If we want H^* as well, we can solve for that too. One of many ways is to realize that, eq. 6.27 can be rearranged to show

$$P_{t+1} = H_t - \frac{H_{t+1}}{R}$$

and given constant H and P ,

$$H^* = P^* \left(\frac{R}{R-1} \right) \tag{6.29}$$

where P^* is eq. 6.26.

6.2.3 Stability of host–parasitoid dynamics

As with differential equation models, we can analyze the stability in the immediate vicinity of equilibria in discrete time models using eigenanalysis of the Jacobian matrix. Although analogous, there are some important differences between continuous and discrete models.

- The Jacobian matrix is comprised of partial derivatives of growth increments (e.g., $\Delta N = N_{t+1} - N_t$), rather than of time derivatives (e.g., $\dot{N}$).
- Interpretation of eigenvalues of the Jacobian reflects the difference between finite rate of increase (λ) and the intrinsic rate of increase (r); e.g., for discrete models, $\lambda = 1$ indicates zero growth.
- Populations with discrete generations can grow too quickly to be stable (e.g., chaos in discrete logistic growth).

In the discrete time case, the Jacobian matrix is the set of *partial derivatives of the discrete growth increment* rather than of the time derivatives used for continuous growth¹². The growth increments are the increments of change over an entire time step, or $\Delta H = H_{t+1} - H_t$ and $\Delta P = P_{t+1} - P_t$. Taking those increments from eqs. 6.26, 6.27, the Jacobian matrix of partial differential equations is

¹² Note dN/dt is also an increment — it is the increment ΔN as $\Delta t \rightarrow 0$, whereas in the discrete case, $\Delta t = 1$.

$$\begin{pmatrix} \frac{\partial \Delta H}{\partial H} & \frac{\partial \Delta H}{\partial P} \\ \frac{\partial \Delta P}{\partial H} & \frac{\partial \Delta P}{\partial P} \end{pmatrix} = \begin{pmatrix} R(1+aP)^{-k} - 1 & -akHR(1+aP)^{-(k+1)} \\ 1 - (1+aP)^{-k} & akH(1+aP)^{-(k+1)} \end{pmatrix} \quad (6.30)$$

To analyze this at the equilibrium, we substitute H^* , P^* for H , P , and perform eigenanalysis. The resulting eigenvalues, λ_1 , λ_2 , then reflect *perturbation growth increments*, the discrete time analogy of the instantaneous perturbation growth rates of continuous models.

Recall that for continuous time, λ_1 is the instantaneous rate of change in a perturbation at equilibrium,

$$\dot{x} = \lambda_1 x, \quad x_t = x_0 e^{\lambda_1 t} \quad (6.31)$$

where x is the perturbation at the equilibrium; if $\lambda_1 < 0$, the perturbation would decline. For discrete growth, we can think of λ_1 as the discrete growth factor of a perturbation at equilibrium,

$$x_{t+1} = \lambda_1 x_t \quad (6.32)$$

where x is the perturbation. Here, we see that x will decline as long as $0 < \lambda_1 < 1$. Second, if $\lambda_1 < 0$, then x_{t+1} changes sign with each time step, and the perturbation oscillates around 0. That is alright, as the magnitude also decreases with each time step, so the perturbation still declines toward zero. However, if $\lambda < -1$, those oscillations grow and the perturbation oscillates permanently. This is directly analogous to the oscillations, or stable limit cycles, we saw in the discrete logistic growth model. Thus, a criterion for stability in discrete growth models is that for all eigenvalues, $-1 < \lambda < 1$.

For discrete models, we also need to be a little more concerned about the imaginary part of the eigenvalues, because they contribute to the magnitude of the oscillations and eigenvalues. We therefore add that the magnitude of $\lambda = a + bi$ is $|\lambda| = \sqrt{a^2 + b^2}$. Thus, the system is stable when $|\lambda| \leq 1$. The magnitude of a complex number is known as its *modulus*. The moduli (*plural* of modulus) of the host–parasite model therefore includes the complex plane, where the real part of each eigenvalue is on the x -axis, and the imaginary part is on the y -axis (Fig. 6.10a). The magnitude or modulus of an eigenvalue is the length of the vector in this plane.

We can illustrate the stability criterion for discrete models in a few ways. The phase plane portrait or time series would illustrate the dynamics directly. However, it is also useful to show the relation between a measure of stability and parameters influencing stability (e.g., Fig. 6.6b). Since aggregation seems so important in the host–parasitoid model, we can show how stability (λ) varies with k . We can thus investigate *how much* aggregation is required for stability [154]. We would anticipate that stability declines ($|\lambda| \rightarrow 1$) as k increases.

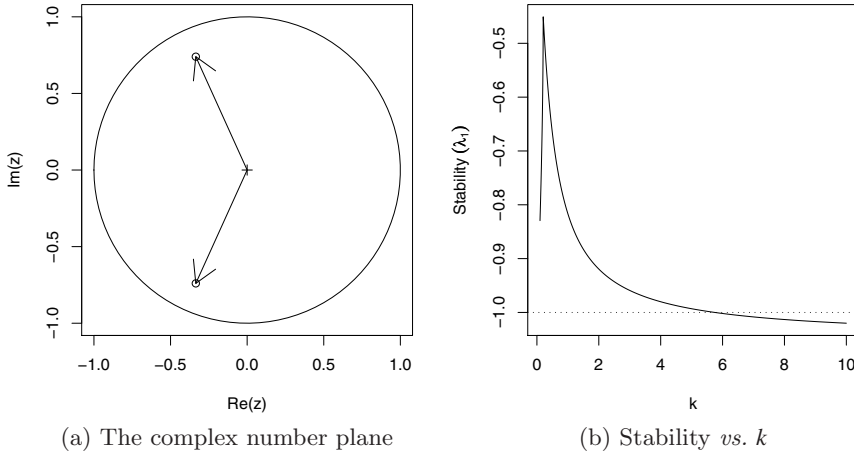


Fig. 6.10: Dynamical stability of a discrete host–parasitoid model with aggregation (a) Region of stability for the rate of change following a small perturbation away from equilibrium. The plus sign “+” is (0,0) and the two small circles are the complex eigenvalues. The length of the vector is the modulus.

Stability of the host–parasitoid model with aggregation (Fig. 6.10b)

We proceed largely as we did for continuous models, first with expressions for, and partial derivatives of, the relevant functions — for discrete models we use $F(N_i)$, where of $N_{i+1} = F(N_i)$.

```
> F.H <- expression(R * H * (1 + a * P/k)^-k - H)
> F.P <- expression(H - H * (1 + a * P/k)^-k - P)
> F.H.H <- D(F.H, "H"); F.H.P <- D(F.H, "P")
> F.P.H <- D(F.P, "H"); F.P.P <- D(F.P, "P")
```

We next specify a sequence of k 's, and for each k , find the equilibria, evaluate the Jacobian, and return the eigenvalues of the Jacobian.

```
> k <- 10^seq(-1, 1, by = 0.01); R <- 3; a <- 0.005
> HPeigs <- sapply(k, function(ki) {
+   k <- ki
+   P <- k * (R^(1/k) - 1)/a
+   H <- P * R/(R - 1)
+   jac <- matrix(c(eval(F.H.H), eval(F.H.P), eval(F.P.H),
+     eval(F.P.P)), nrow = 2, byrow = TRUE)
+   eigen(jac)[["values"]]
+ })
```

Last, we plot the eigenvalue with the greatest absolute magnitude, and retain the sign of the real part, λ vs. k .

```
> modmaxs <- apply(HPeigs, 2, function(lambdas) {
+   i <- which.max(Mod(lambdas))
+   sign(Re(lambdas[i])) * Mod(lambdas[i])
+ })
> plot(k, modmaxs, type = "l", ylab = quote("Stability " *
+   (lambda[1])))
> abline(h = -1, lty = 3)
```

Graphing eigenvalues in the complex number plane (Fig. 6.10a)

It is typically important to evaluate the modulus, or magnitude, of the eigenvalues of a Jacobian matrix for a discrete model. First we set up the unit circle which will define the stability region in the complex number plane.

```
> th <- seq(-pi, pi, len = 100)
> z <- exp((0+1i) * th)
```

We then plot the circle and add the eigenvalues for our smallest k ;

```
> par(pty = "s")
> plot(z, type = "l")
> points(0, 0, pch = 3)
> points(HPeigs[, 100])
> arrows(x0 = c(0, 0), y0 = c(0, 0), x1 = Re(HPeigs[, 100]),
+       y1 = Im(HPeigs[, 100]))
```

The length of the arrows are the moduli, $|\lambda|$.

Dynamics of the May host–parasitoid model

Here we simply play with May's model. The following generates a phase plane diagram of the dynamics, although it is not shown. We set the duration of the time series, the model parameters, and create an empty data frame to hold the results.

```
> time <- 20
> R <- 3
> a <- 0.005
> k <- 0.6
> HP2s <- data.frame(Hs <- numeric(time), Ps <- numeric(time))
```

Next we calculate the equilibrium, and use a nearby point for the initial abundances.

```
> P2st <- k * (R^(1/k) - 1)/a
> H2st <- P2st * R/(R - 1)
> P2st
```

```
[1] 628.8
```

```
> H2st
```

```
[1] 943.2
```

```
> HP2s[1, ] <- c(1000, 500)
```

We then project the dynamics, one generation at a time, for each time step.

```
> for (t in 1:(time - 1)) HP2s[t + 1, ] <- {
+   H <- R * HP2s[t, 1] * (1 + a * HP2s[t, 2]/k)^(-k)
+   P <- HP2s[t, 1] - HP2s[t, 1] * (1 + a * HP2s[t, 2]/k)^(-k)
+   c(H, P)
+ }
```

Last we plot the trajectory, placing a point at the equilibrium, and using arrows to highlight the direction and magnitude of the increment of change per time step.

```
> plot(HP2s[, 1], HP2s[, 2], type = "l", xlab = "H", ylab = "P")
> points(H2st, P2st)
> arrows(HP2s[-time, 1], HP2s[-time, 2], HP2s[-1, 1], HP2s[-1,
+      2], length = 0.05)
```

6.3 Disease

Here we discuss epidemiological disease models. Pathogens cause diseases, and are typically defined as microorganisms (fungi, bacteria, and viruses) with some host specificity, and which undergo population growth within the host.

Our simplest models of disease are funny, in that they don't model pathogens (the enemy) at all. These famous models, by Kermack and McCormick [91], keep track of different types of hosts, primarily those with and without disease symptoms. That makes them *epidemiological models*. Specifically, these are SIR models [91] that model all N hosts by keeping track of

Susceptible hosts Individuals which are not infected, but could become infected,

Infected hosts Individuals which are already infected, and

Resistant hosts Individuals which are resistant to the disease, typically assumed to have built up an immunity through past exposure,

where $N = S + I + R$. It is important to note that N , S , I , and R are *densities*. That is, we track numbers of individuals in a fixed area. This is important because it has direct consequences for the spread, or transmission, of disease [136].

Disease spreads from infected individuals to susceptible individuals. The rate depends to some extent on the number or alternatively, on the fraction of the population that is infected. Resistant individuals are not typically considered vectors of the pathogens, and so increased abundance of resistant individuals slow the transmission rate by diluting the other two groups.

A good starting place is a simple SIR model for a population of constant size, with no immigration or emigration [48, 91].

$$\frac{dS}{dt} = -\beta IS \quad (6.33)$$

$$\frac{dI}{dt} = \beta IS - \gamma I \quad (6.34)$$

$$\frac{dR}{dt} = \gamma I \quad (6.35)$$

Density-dependent SIR model

Here we create the function for the system of ODE's in eq. 6.33.

```
> SIR <- function(t, y, p) {
+   {
+     S <- y[1]
+     I <- y[2]
+     R <- y[3]
+   }
+   with(as.list(p), {
+     dS.dt <- -B * I * S
+     dI.dt <- B * I * S - g * I
+     dR.dt <- g * I
+     return(list(c(dS.dt, dI.dt, dR.dt)))
+   })
+ }
```

In this model, the *transmission coefficient*, β , describes the instantaneous rate at which the number of infected hosts increases per infected individual. It is directly analogous to the attack rate of type I Lotka–Volterra predator–prey models. Recall that it is based on the law of mass action, borrowed from physics and chemistry. It assumes that the rate at which events occur (new infections) is due to complete and random mixing of the reactants (S , I), and the rate at which the reactants collide and react can be described by a single constant, β . As density of either type of molecule increases, so too does the rate of interaction. In Lotka–Volterra predation, we referred to βS as a linear functional response; here we refer to βS as the *transmission function* and in particular we call it a *mass action* or *density-dependent transmission function*. The *transmission rate* is the instantaneous rate for the number of *new infections* or cases per unit time [136].

Resistant individuals might be resistant for one of two reasons. They may die, or they may develop immunities. In either case, we assume they cannot catch the disease again, nor spread the disease. As this model assumes a constant population size, we continue to count all R individuals, regardless of whether they become immune or die.

The individuals become resistant to this disease at the constant per capita rate, γ . The rate γ is also the inverse of the mean residence time, or *duration*, of the disease¹³.

Disease *incidence* is the number of new infections or cases occurring over a defined time interval. This definition makes incidence a discrete-time version of transmission rate. *Prevalence* is the fraction of the population that is infected I/N .

A common question in disease ecology is to ask under what conditions will an outbreak occur? Another way of asking that is to ask what conditions cause

¹³ This is an interesting phenomenon of exponential processes — the mean time associated with the process is equal to the inverse of the rate. This is analogous to turnover time or residence time for a molecule of water in a lake.

$\dot{I} > 0$. We can set $dI/dt > 0$ and solve for something interesting about what is required for an outbreak to occur.

$$\begin{aligned} 0 &< \beta IS - \gamma I \\ \frac{\gamma}{\beta} &< S \end{aligned} \tag{6.36}$$

What does this tell us? First, because we could divide through by I , it means that if no one is infected, then an outbreak can't happen — it is the usual, but typically unstable equilibrium at 0. Second, it tells us that an outbreak will occur if the absolute density of susceptibles¹⁴ is greater than γ/β . If we consider the pre-outbreak situation where $S \approx N$, then simply making the population size (and density) low enough can halt the spread of disease. This is why outbreaks tend to occur in high density populations, such as agricultural hosts (e.g., cattle), or historically in urban human populations, or in schools.

Vaccinations are a way to reduce S without reducing N . If a sufficient number of individuals in the population are vaccinated to reduce S below γ/β , this tends to protect the unvaccinated individuals as well.

Another common representation of this is called the “force of infection” or “basic reproductive rate of the disease.” If we assume that in a large population $S \approx N$, then rearranging eq. 6.36 gives us

$$R_0 = \frac{\beta N}{\gamma} \tag{6.37}$$

where R_0 is the basic reproductive rate of the disease. If $R_0 > 1$, then an outbreak (i.e. disease growth) is plausible. This is analogous to the finite rate of increase of a population where $\lambda > 1$.

Simple SIR dynamics (Fig. 6.11)

Here we model the outbreak of a nonlethal disease (e.g., a new cold virus in winter on a university campus). We assume that the disease is neither live-threatening, and nor is anyone resistant, thus $R_{t=0} = 0$. We can investigate the SIR model by pretending that, as is often the case, we begin with a largely uninfected population and $t = 0$, so $I_0 = 1$ and $S_0 \approx N$. We first set parameters.

```
> N <- 10^4
> I <- R <- 1
> S <- N - I - R
> parms <- c(B = 0.01, g = 4)
```

We next integrate for three months.

```
> months <- seq(0, 3, by = 0.01)
> require(deSolve)
> SIR.out <- data.frame(ode(c(S, I, R), months, SIR, parms))

> matplot(months, SIR.out[, -1], type = "l", lty = 1:3, col = 1)
> legend("right", c("R", "I", "S"), lty = 3:1, col = 3:1, bty = "n")
```

¹⁴ S is the absolute density, whereas S/N is the relative density.

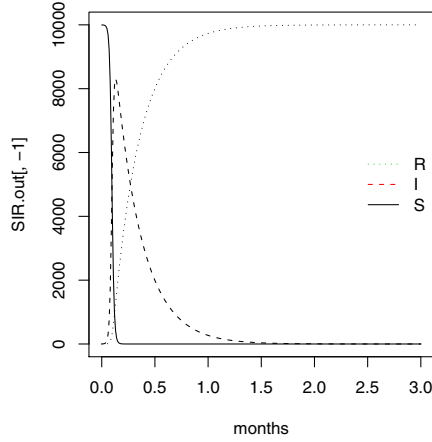


Fig. 6.11: Epidemic with the simplest SIR model. Assumes constant population size.

6.3.1 SIR with frequency-dependent transmission

It is important at this point to reiterate a point we made above — *these conclusions apply when S , I , and R are densities* [136]. If you increase population size but also the area associated with the population, then you have not changed density. If population size only increases, but density is constant, then interaction frequency does not increase. Some populations may increase in density as they increase in size, but some may not. Mass action dynamics are the same as type I functional response as predators — there is a constant linear increase in per capita infection rate as the number of susceptible hosts increases.

In addition to mass action (a.k.a. density dependent) transmission, investigators have used other forms of density dependence. One the most common is typically known as *frequency-dependent* transmission, where

$$\frac{dS}{dt} = -\beta \frac{SI}{N} \quad (6.38)$$

$$\frac{dI}{dt} = \beta \frac{SI}{N} - \gamma I \quad (6.39)$$

$$\frac{dI}{dt} = \gamma I. \quad (6.40)$$

Frequency-dependent SIR model

Here we create the function for the system of ODEs in eq. 6.33.

```
> SIRf <- function(t, y, p) {
+   {
+     S <- y[1]
+     I <- y[2]
+     R <- y[3]
+     N <- S + I + R
+   }
+   with(as.list(p), {
+     dS.dt <- -B * I * S/N
+     dI.dt <- B * I * S/N - g * I
+     dR.dt <- g * I
+     return(list(c(dS.dt, dI.dt, dR.dt)))
+   })
+ }
```

The proper form of the transmission function depends on the mode of transmission [136]. Imagine two people are on an elevator, one sick (infected), and one healthy but susceptible, and then the sick individual sneezes [48]. This results in a particular probability, β , that the susceptible individual gets infected. Now imagine resistant individuals get on the elevator — does will adding resistant individuals change the probability that the susceptible individual gets infected? Note what has and has not changed. First, with the addition of a resistant individual, N has increased, and prevalence, I/N , has decreased. However, the densities of I and S remain the same (1 per elevator). What might happen? There are at least two possible outcomes:

1. If sufficient amounts of the virus spread evenly throughout the elevator, adding a resistant individual does *not* change the probability of the susceptible becoming sick, and the rate of spread will remain dependent on the densities of I and S — the rate will not vary with declining prevalence.
2. If only the immediate neighbor gets exposed to the pathogen, then the probability that the neighbor is susceptible declines with increasing R , and thus the rate of spread *will* decline with declining prevalence.

It is fairly easy to imagine different scenarios, and it is very important to justify the form of the function.

Density-dependent transmission (Fig. 6.12) is independent of the number of resistant individuals; having higher density of infected individuals or more susceptible individuals always enhances transmission rate, assuming both $I, S > 0$. In contrast, frequency-dependent transmission does depend on the density of resistant (living) individuals because they can “get in the way” of transmission (consider our elevator example above). That is, they reduce to probability of infected individuals coming into contact with susceptible hosts¹⁵. That is, there is a greater probability that the immediate neighbor of an infected host is already infected, and so cannot become a new infection or case. Similarly, having more

¹⁵ Note $N = S + I + R$, and that increasing R cause decreasing SI/N .

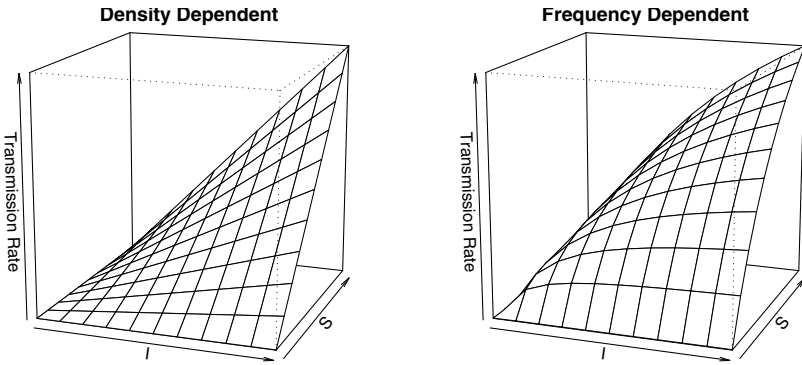


Fig. 6.12: The rate of transmission may depend only on the linear dependence of mass action, or may depend curvilinearly on the prevalence, the frequency of infected individuals in the population.

susceptible hosts makes it more likely the immediate neighbor of a susceptible host is another susceptible host and not a source of infection.

Transmission models (Fig. 6.12)

Here we plot density-dependent and frequency-dependent transmission rates, as functions of S and I . We rescale the transmission coefficient appropriately ($\beta_F = N_{\max}\beta_D$) [136].

```
> R <- 0
> S <- I <- 1000
> Ss <- Is <- seq(1, S, length = 11)
> N <- S + I + R
> betaD <- 0.1
> betaF <- betaD * N
```

We use `sapply` to calculate the transmission functions for each combination of the values of I and S .

```
> mat1 <- sapply(Is, function(i) betaD * i * Ss)
> mat2 <- sapply(Is, function(i) betaF * i * Ss/(i + Ss + R))
```

Now we plot these matrices.

```
> layout(matrix(1:2, nr = 1))
> persp(mat1, theta = 20, phi = 15, r = 10, zlim = c(0, betaD *
+   S * I), main = "Density Dependent", xlab = "I", ylab = "S",
+   zlab = "Transmission Rate")
> persp(mat2, theta = 20, phi = 15, r = 10, zlim = c(0, betaF *
+   S * I/N), main = "Frequency Dependent", xlab = "I", ylab = "S",
+   zlab = "Transmission Rate")
```

What does frequency–dependent transmission imply about dynamics? Let's solve for $dI/dt > 0$.

$$\begin{aligned} 0 &< \beta \frac{SI}{N} - \gamma I \\ \gamma &< \beta \frac{S}{N}. \end{aligned} \tag{6.41}$$

As we did above, let's consider the pre-outbreak situation where $S \approx N$, so that $S/N \approx 1$. In that case, the basic reproductive rate is $R_0 = \beta/\gamma$, which is *independent of* N . An outbreak will occur as long as $\beta > \gamma$, regardless of population density. This is in direct contrast to the density–dependent transmission model (eqs. 6.38, 6.40), where outbreak could be prevented if we simply reduce the population, N , to a sufficiently low density. Both modes of transmission are observed in human and non-human populations, so it is important to understand how the disease is spread in order to predict its dynamics.

Another interesting phenomenon with frequency–dependent transmission is that prevalence (I/N) can decline with increasing population size (Fig. 6.13). Two phenomena contribute to this pattern. First, outbreak in a completely susceptible population typically begins with a single individual, and so initial prevalence is always $I/N = 1/N$. Second, as a consequence of this, the transmission rate is lower in larger populations because $\beta SI/N$ is small. As a consequence, prevalence remains low for a relatively long time. In a seasonal population, most individuals in large populations remain uninfected after four months. Depending upon the organism, this could be long enough to reproduce. In contrast, a density–dependent model typically shows the opposite, pattern, with more rapid, extreme outbreaks and greater prevalence in more dense populations (Fig. 6.13).

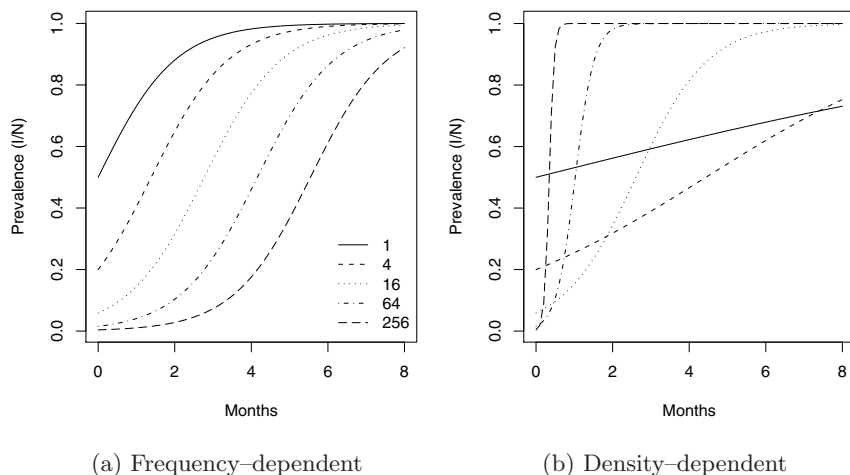


Fig. 6.13: Prevalence (I/N) vs. population density. With frequency-dependent transmission, (a), prevalence may decrease with population density. In contrast, with density-dependent transmission, (b), prevalence may increase with density.

SIR dynamics with frequency-dependent transmission (Fig. 6.13)

Here we demonstrate that prevalence can decline with increasing population size in a frequency-dependent disease (e.g., a smut on plant [3]). Let us assume that resistance cannot be acquired, so $\gamma = 0$, and $R = 0$. Here we can investigate the SIR model by pretending that, as is often the case, we begin with a largely uninfected population and $t = 0$, so $I_0 = 1$ and $S_0 \approx N$. We first set parameters.

```
> S <- 4^(0:4)
> I <- 1
> parmsf <- c(B = 1, g = 0)
> parmsd <- c(B = 1/16, g = 0)
```

We next integrate for six months, letting $R = S/2$.

```
> Months <- seq(0, 8, by = 0.1)
> outd <- sapply(S, function(s) {
+   out <- ode(c(s, I, R), Months, SIR, parmsd)
+   out[, 3]/apply(out[, 2:4], 1, sum)
+ })
> outf <- sapply(S, function(s) {
+   out <- ode(c(s, I, R), Months, SIRf, parmsf)
+   out[, 3]/apply(out[, 2:4], 1, sum)
+ })
```

Last, we make the figures.

```
> matplot(Months, outd, type = "l", col = 1, ylab = "Prevalence (I/N)")
> matplot(Months, outf, type = "l", col = 1, ylab = "Prevalence (I/N)")
> legend("bottomright", legend = S, lty = 1:length(S), bty = "n")
```

6.3.2 SIR with population dynamics

(The following sections rely on code.)

The above model assumes a constant population size — sort of. Recall that the “resistant group” could consist of those that acquire the ultimate immunity, death. In any event, we could make a more complex model that includes population growth and death unrelated to disease. Here we add births, b , potentially by all types, sick or not ($S + I + R$), and we assume that the newborns are susceptible only. We also added a mortality term to each group (mS, mI, mR).

$$\frac{dS}{dt} = b(S + I + R) - \beta SI - mS \quad (6.42)$$

$$\frac{dI}{dt} = \beta SI - \gamma I - mI \quad (6.43)$$

$$\frac{dR}{dt} = \gamma I - mR \quad (6.44)$$

Note that the births add only to the susceptible group, whereas density independent mortality subtracts from each group.

Disease model with population growth

Here we create the function for the system of ODE’s in eq. 6.42.

```
> SIRbd <- function(t, y, p) {
+   S <- y[1]
+   I <- y[2]
+   R <- y[3]
+   with(as.list(p), {
+     dS.dt <- b * (S + I + R) - B * I * S - m * S
+     dI.dt <- B * I * S - g * I - m * I
+     dR.dt <- g * I - m * R
+     return(list(c(dS.dt, dI.dt, dR.dt)))
+   })
+ }
```

Let’s start to work with this model — that frequently means making simplifying assumptions. We might start by assuming that if infected and resistant individuals can contribute to offspring, then the disease is relatively benign. Therefore, we can assume that mortality is the same for all groups ($m_i = m$). Last, let us assume (again) a constant population size. This means that birth rate equals mortality or $b = m$.

Now imagine a large city, with say, a million people. Let’s then assume that we start of with a population of virtually all susceptible people, but we introduce a single infected person.

```
> N <- 10^6
> R <- 0
> I <- 1
> S <- N - I - R
```


Let us further pretend that the disease runs its course over about 10–14 days. Recall that γ (“gamma”) is the inverse of the *duration* of the disease.

```
> g <- 1/(13/365)
```

Given a constant population size and exponential growth, then the average life span is the inverse of the birth rate. Let us pretend that the average life span is 50 years.

```
> b <- 1/50
```

For this model, the force of infection turns out to be $R_0 = 1 + 1/(b + \alpha)$, where α is the average age at onset of the disease [48]. We can therefore estimate β from all the other parameters, including population size, average life span, average age at onset, and the average duration of the disease. For instance, imagine that we have a disease of children, where the average onset of disease is 5 y, so we have

```
> age <- 5
> R0 <- 1 + 1/(b * age)
```

so β becomes

```
> B <- R0 * (g + b)/N
```

Finally, we can integrate the population and its states. We create a named vector of parameters, and decide on the time interval.

```
> parms <- c(B = B, g = g, b = b, m = b)
> years <- seq(0, 30, by = 0.1)
```

It turns out that because of the relatively extreme dynamics (Fig. 6.14), we want to tell the ODE solver to take baby steps, so as to properly capture the dynamics — we use the `hmax` argument to make sure that the maximum step it takes is relatively small.

```
> SIRbd.out <- data.frame(ode(c(S = S, I = I, R = R), years,
+   SIRbd, parms, hmax = 0.01))

> matplot(SIRbd.out[, 1], sqrt(SIRbd.out[, -1]), type = "l",
+   col = 1, lty = 1:3, ylab = "sqrt(No. of Individuals)",
+   xlab = "Years")
> legend("right", c("S", "I", "R"), lty = 1:3, bty = "n")
```

Note that the population quickly becomes resistant (Fig. 6.14). Note also that we have oscillations, albeit damped oscillations. An analytical treatment of the model, including eigenanalysis of the Jacobian matrix could show us precisely the predicted periodicity [48]. It depends on the the age at onset, and the duration of the disease.

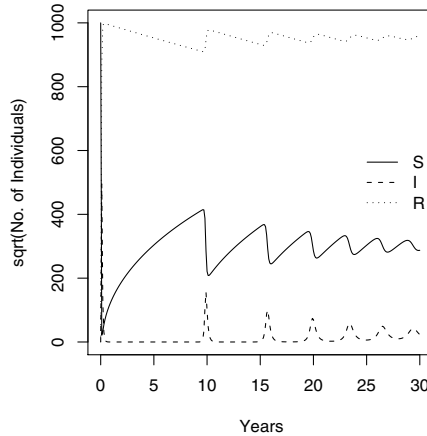


Fig. 6.14: Epidemic for a nonlethal disease, with an SIR model which includes births and deaths, and a constant population size.

6.3.3 Modeling data from Bombay

Here we try our hand at fitting the SIR model to some data. Kermack and McCormick [91] provided data on the number of plague deaths per week in Bombay¹⁶ in 1905–06. We first enter them and look at them¹⁷.

```
> data(ross)
> plot(CumulativeDeaths ~ Week, data = ross)
```

As with most such enterprises, we wish we knew far more than we do about the which depends on fleas, rodents, humans, and *Yersinia pestis* on which the dynamics depend. To squeeze this real-life scenario into a model with a small number of parameters requires a few assumptions.

A good starting place is a simple SIR model for a population of constant size (eq. 6.33) [48,91].

Optimization

We next want to let R find the most accurate estimates of our model parameters β , γ . The best and most accessible reference for this is Bolker [13]. Please read the Appendix B.11 for more explanation regarding optimization and objective functions.

Now we create the objective function. An objective function compares a model to data, and calculates a measure of fit (e.g., residual sum of squares, likelihood). Our objective function will calculate the likelihood¹⁸ of particular

¹⁶ Bombay is the coastal city now known as Mumbai, and is the capital of Maharashtra; it is one of the largest cities in the world.

¹⁷ Data provided kindly by S.P. Ellner

¹⁸ *Likelihood* is the probability of data, given a particular model and its parameters.

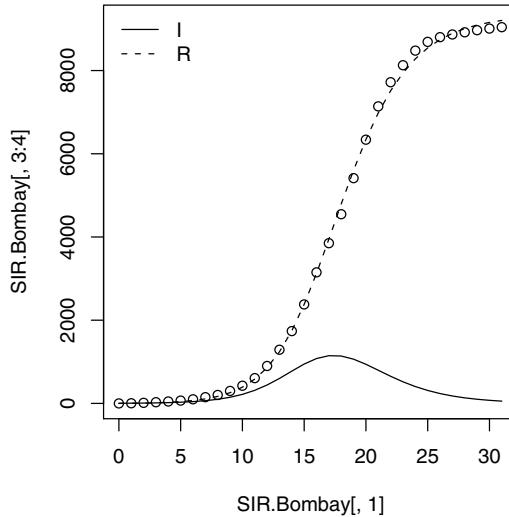


Fig. 6.15: Cumulative deaths for plague, in Bombay, India, 1905–1906 (raw data and fitted model, as described in this section).

values for SIR model parameters. These parameters include γ and β of the SIR model. The parameters will also include two other unknowns, (i) N , total relevant population size of Bombay at the time (1905–1906), and (ii) I_0 , initial size of the infected population at our $t = 0$. A survey of easily accessed census data suggests the population at the time was in the neighborhood of $\sim 10^6$ individuals. We also might assume that I_0 would be a very small number, perhaps ~ 1 in principle.

Some details about our objective function:

1. Arguments are transformed parameters (this allows the optimizer to work on the logit¹⁹ and log scales).
2. Transformed parameters are backtransformed to the scale of the model.
3. Parameters are used to integrate the ODEs using `ode`, retaining only the resistant (i.e. dead) group (the fourth column of the output); this provides the *predicted values* given the parameters and the time from onset, and the standard deviation of the residuals around the predicted values.
4. It returns the negative log-likelihood of the data, given the parameter values.

Here is our objective function. Note that its last value is the negative sum of the log-probabilities of the data (given a particular realization of the model).

```
> sirLL = function(logit.B, logit.g, log.N, log.I0) {
+   parms <- c(B = plogis(logit.B), g = plogis(logit.g))
+   x0 <- c(S = exp(log.N), I = exp(log.I0), R = 0)
+   Rs <- ode(y = x0, ross$Week, SIR, parms, hmax = 0.01)[,
```

¹⁹ A logit is the transformation of a proportion which will linearize the logistic curve, $\text{logit}(p) = \log(p/(1-p))$.

```
+      4J
+      SD <- sqrt(sum((ross$CumulativeDeaths - Rs)^2)/length(ross$Week))
+      -sum(dnorm(ross$CumulativeDeaths, mean = Rs, sd = SD,
+      log = TRUE))
+ }
}
```

We then use this function, `sirLL`, to find the likelihood of the best parameters. The `mle2` function in the `bbmle` library²⁰ will minimize the negative log-likelihood generated by `sirLL`, and return values for the parameters of interest.

We will use a robust, but relatively slow method called Nelder-Mead (it is the default). We supply `mle2` with the objective function and a list of initial parameter values. This can take a few minutes.

```
> require(bbmle)
> fit <- mle2(sirLL, start = list(logit.B = qlogis(1e-05),
+   logit.g = qlogis(0.2), log.N = log(1e+06), log.I0 = log(1)),
+   method = "Nelder-Mead")

> summary(fit)
```

Maximum likelihood estimation

Call:

```
mle2(minuslogl = sirLL, start = list(logit.B = qlogis(1e-05),
  logit.g = qlogis(0.2), log.N = log(1e+06), log.I0 = log(1)),
  method = "Nelder-Mead")
```

Coefficients:

	Estimate	Std. Error	z value	Pr(z)
logit.B	-9.4499	0.0250	-377.39	<2e-16
logit.g	1.0180	0.0663	15.34	<2e-16
log.N	9.5998	0.0181	530.15	<2e-16
log.I0	1.2183	0.1386	8.79	<2e-16

-2 log L: 389.2

This gets us some parameter estimates, but subsequent attempts to actually get confidence intervals failed. This occurs frequently when we ask the computer to estimate too many, often correlated, parameters for a given data set. Therefore, we have to make assumptions regarding selected parameters. Let us assume for the time being that the two variable estimates are correct, that the population size of the vulnerable population was 10 and the number of infections at the onset of the outbreak was 1. We will hold these constant and ask R to refit the model, using the default method.

```
> fit2 <- mle2(sirLL, start = as.list(coef(fit)),
+   fixed = list(log.N = coef(fit)[3], log.I0 = coef(fit)[4]),
+   method = "Nelder-Mead")
```

²⁰ You will need to load the `bbmle` package from a convenient mirror, unless someone has already done this for the computer you are using. See the Appendix for details about packages (A.3) and optimization in R (B.11).

```
> summary(fit2)
```

Maximum likelihood estimation

Call:
mle2(minuslogl=sirLL, start = as.list(coef(fit)), method = "Nelder-Mead",
fixed = list(log.N = coef(fit)[3], log.I0 = coef(fit)[4]))

Coefficients:

	Estimate	Std. Error	z value	Pr(z)
logit.B	-9.44971	0.00632	-1495.4	<2e-16
logit.g	1.01840	0.03423	29.8	<2e-16

-2 log L: 389.2

Next we want to find confidence intervals for β and γ . This can take *several* minutes, but results in a likelihood profile for these parameters, which show the confidence regions for these parameters (Fig. 6.16).

```
> pr2 <- profile(fit2)
> par(mar = c(5, 4, 4, 1))
> plot(pr2)
```

We see that the confidence intervals for the transformed variables provide estimates of our confidence in these parameters.

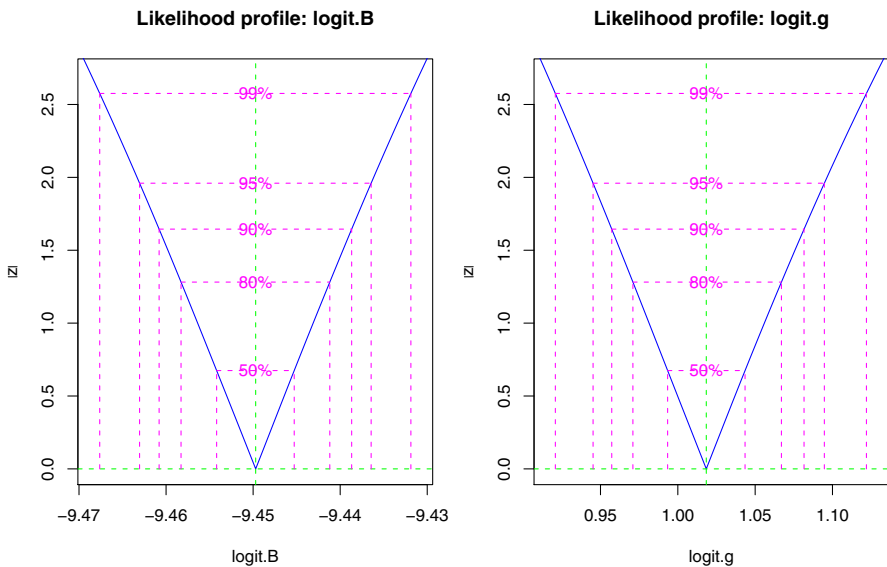


Fig. 6.16: Likelihood profile plots, indicating confidence intervals on transformed SIR model parameters.

Last we get to plot our curve with the data. We first backtransform the coefficients of the objective function.

```
> p <- as.numeric(c(plogis(coef(fit2)[1:2]), exp(coef(fit2)[3:4])))
> p
[1] 7.871e-05 7.347e-01 1.476e+04 3.381e+00
```

We then get ready to integrate the disease dynamics over this time period.

```
> inits <- c(S = p[3], I = p[4], R = 0)
> params <- c(B = p[1], g = p[2])
> SIR.Bombay <- data.frame(ode(inits, ross$Week, SIR, params))
```

Last, we plot the model and the data (Fig. 6.15).

```
> matplot(SIR.Bombay[, 1], SIR.Bombay[, 3:4], type = "l", col = 1)
> points(ross$Week, ross$CumulativeDeaths)
> legend("topleft", c("I", "R"), lty = 1:2, bty = "n")
```

So, what does this mean (Fig. 6.15)? We might check what these values mean, against what we know about the reality. Our model predicts that logit of γ was a confidence interval,

```
> (CIs <- confint(pr2))

          2.5 % 97.5 %
logit.B -9.4630 -9.436
logit.g  0.9452  1.095
```

This corresponds to a confidence interval for γ of

```
> (gs <- as.numeric(plogis(CIs[2, ])))
[1] 0.7201 0.7493
```

Recall that the duration of the disease in the host is $1/\gamma$. Therefore, our model predicts a confidence for the duration (in weeks) of

```
> 7 * 1/gs
[1] 9.720 9.342
```

Thus, based on this analysis, the duration of the disease is right around 9.5 days. This seems to agree with what we know about the biology of the Bubonic plague. Its duration, in a human host, is typically thought to last 4–10 days.

6.4 Summary

This chapter has skimmed the surface of a tremendous amount of territory. Some of the things we've learned include the notion that predator–prey relations can unstable (Lotka–Volterra predator–prey), and that additional biology stabilize or destabilize dynamics (negative density dependence, type II and III functional responses); increasing resources does not always help those we intend to help

(paradox of enrichment). We learned that space, and the spatial distribution of hosts and their parasitoids matter (host–parasitoid dynamics) to population dynamics. We also learned that disease outbreaks are expected, under some conditions and not others, and that the mode of disease transmission matters, and can be modeled in a variety of ways.

Problems

6.1. Lotka–Volterra Predator–prey Model

- Write down the two species Lotka–Volterra predator–prey model.
- Describe how Fig. 6.4 illustrates neutral oscillatory dynamics.
- What are the units of the predator–prey model coefficients b , a , e , and s ? How do we interpret them?

6.2. Rosenzweig–MacArthur Predator–prey Model

- Write down the two species Rosenzweig–MacArthur predator–prey model.
- How do we interpret b , K , w , D , e and s ? What are their units?
- What is the value of the functional response when $H = D$? Explain how this result provides the meaning behind the name we use for D , the *half saturation* constant.
- For each point A–D in Fig. 6.5, determine whether the growth rate for the predator and the herbivore are zero, positive, or negative.
- In what way is the Rosenzweig–MacArthur predator isocline (Fig. 6.5) similar to the Lotka–Volterra model? It also differs from the Lotka–Volterra isocline—explain the ecological interpretation of D in the type II functional response and its consequence for this isocline.
- Explain the interpretation of real and imaginary parts of the eigenvalues for this parameterization of the Rosenzweig–MacArthur predator–prey model.
- In what ways does Fig. 6.6a match the interpretation of the eigenanalysis of this model?
- Examine the prey isoclines in Fig. 6.6a. How you can tell what the carrying capacities of the prey are?
- What do the above eigenanalyses tell us about how the stability of the predator–prey interaction varies with the carrying capacity of the prey?
- Consult Fig. 6.6a. What is the relation between the carrying capacity of the prey and the magnitude of the oscillations? What is the relation between the carrying capacity of the prey and the minimum population size? What does this interpretation imply about natural ecosystems?

6.3. Effects of dispersion on host–parasitoid dynamics

- Demonstrate the effects of aggregation on host–parasitoid dynamics. Specifically, vary the magnitude of k to find the effects on stability.
- Demonstrate the effects of a on stability.
- Demonstrate the effects of R on stability.

6.4. Effects of age at onset and disease duration on outbreak periodicity

- (a) Create three simulations showing how diseases of different durations influence the periodicity of the outbreaks.
- (b) Create three simulations showing how the age at onset for different diseases influence the periodicity of the outbreaks.
- (c) Consider which factor is more important in influencing outbreak interval. How do you measure the interval? What criteria would you use to determine “importance”? How do the realities of age and duration influence your selection of criteria? Write a short essay that asserts a thesis, and then provides support based on this exercise.